AStA Adv Stat Anal (2009) 93: 307–333
DOI 10.1007/s10182-009-0113-6

ORIGINAL PAPER

A Bayesian latent variable approach to functional principal components analysis with binary and count data

Angelika van der Linde

Received: 25 March 2009 / Accepted: 24 July 2009 / Published online: 18 August 2009

Abstract Recently, van der Linde (Comput. Stat. Data Anal. 53:517–533, 2008) proposed a variational algorithm to obtain approximate Bayesian inference in functional principal components analysis (FPCA), where the functions were observed with Gaussian noise. Generalized FPCA under different noise models with sparse longitudinal data was developed by Hall et al. (J. R. Stat. Soc. B 70:703–723, 2008), but no Bayesian approach is available yet. It is demonstrated that an adapted version of the variational algorithm can be applied to obtain a Bayesian FPCA for canonical parameter functions, particularly log-intensity functions given Poisson count data or logit-probability functions given binary observations. To this end a second order Taylor expansion of the log-likelihood, that is, a working Gaussian distribution and hence another step of approximation, is used. Although the approach is conceptually straightforward, difficulties can arise in practical applications depending on the accuracy of the approximation and the information in the data. A modified algorithm is introduced generally for one-parameter exponential families and exemplified for binary and count data. Conditions for its successful application are discussed and illustrated using simulated data sets. Also an application with real data is presented.

Keywords Probabilistic PCA · Logistic PCA · Factor analysis · Exponential family · Variational algorithm · Working observations · Splines

1 Introduction

Imagine that for each individual in a group the occurrence or frequency of an event is recorded over a certain period of time, for example, in a longitudinal survey of patients (Hall et al. 2008). Another example is given by Ramsay and Silverman (2002)

A. van der Linde (✉)
FB03: Institute of Statistics, University of Bremen, P.O. Box 330 440, 28334 Bremen, Germany
e-mail: avdl@math.uni-bremen.de

who present a case study in criminology where for each of about 400 men the annual number of arrests over a 25-years period of their lives is analyzed. As yet another example Smith (1987) discusses modelling rainfall data at different sites with Bernoulli processes. Thus for each individual there is a function over time describing the probability or intensity of occurrence of the event. Often it is of interest to study the variation of curves within the group and to identify characteristic modes of variation. This problem is addressed in functional principal components analysis (FPCA). There is a vast literature on FPCA for curves observed with Gaussian noise (see the review given by van der Linde 2008) but there are only few attempts to address the problem with binary or count data.

Mathematically, binary or count data are regarded as realizations of random variables following a distribution from a one-parameter exponential family, and the function over time characterizing an individual is chosen to be either the mean function or the canonical parameter function, that is, a link transformed mean function. The problem then is to apply FPCA to these functions and to subsequently reconstruct the individual functions with possibly only sparse data available.

Behseta et al. (2005) suggest a Bayesian approach for spiky curves, requiring that sufficient data is available to estimate each spiky curve separately (with free knot splines). The estimates then are treated like Normally distributed observations of the unknown functions. In general, PCA based on individually estimated curves, called the *naive approach* by James et al. (2000), does not take advantage of the similarity of curves typically present in a set of (non-spiky) curves, and for sparse data the naive approach is not applicable. Hall et al. (2008) propose a procedure for data from one-parameter exponential families in which the canonical parameter functions are modeled as trajectories of a Gaussian latent process. Exploiting the link transformation the covariance function of this process is estimated nonparametrically and its Karhunen–Loève expansion in eigenfunctions is used in further inference, particularly for estimating principal component scores in order to recover the individual canonical parameter functions. The assessment of errors of the estimates—if required—is based on bootstrapping.

Following James et al. (2000) it can be argued that rather than estimating a full covariance function it may be easier and more efficient to estimate directly only the first few eigenfunctions in a generative model. In a Bayesian approach to (functional) PCA, the generative model is attractive, and along these lines a Bayesian FPCA for Gaussian data was devised by van der Linde (2008). In this paper, the approach is generalized to canonical parameter functions of one-parameter exponential families. The main problem in extending the method based on a variational algorithm to obtain approximate posterior distributions is that a tractable lower bound of the log-likelihood function allowing for inference in closed form is required; it can be provided for binary data, but is not available for count data.

(Bayesian) PCA for exponential families has been of interest also apart from a functional context, especially in the machine learning community, and several proposals were published almost simultaneously. Kabán and Bingham (2008) decomposed the means (success probabilities) of Bernoulli-distributed variables into a convex combination of a priori Beta-distributed latent factors, and in this special case derived a lower bound of the likelihood function which can be used in a variational

algorithm. Dimension reduction is less frequently obtained using the mean parameterization with constraints on the range of values than using the canonical parameterization. Collins et al. (2002) introduced a principled approach to PCA in one-parameter exponential families advocating an approximation of canonical parameter vectors in a lower dimensional linear subspace. In terms of a generative model, their approach is analogous to the extension of Gaussian regression to Generalized Linear Models (GLMs) with maximum likelihood estimation. Wedel and Kamakura (2001) suggested the same generative model for univariate exponential families, but allowed for latent factors with densities in another univariate exponential family. They investigated simulated likelihood estimation for further inference.

An influential paper that demonstrated the applicability of a variational algorithm for approximate Bayesian inference about canonical parameters given binary data was that by Jaakkola and Jordan (1997). Their approach is built on a convexity inequality for the logistic function and hence cannot be extended to other exponential families. It was applied to probabilistic PCA by Tipping (1999) and also used to derive an algorithm in simple closed form for non-probabilistic PCA by Schein et al. (2003). The main problem in generalizing from Bernoulli distributions to densities from other exponential families as well as to the multivariate multinomial logit model is the lack of a general lower bound of the log-likelihood. A way out, tried successfully for the multinomial model by Braun and McAuliffe (2008), is to give up the idea of an exact lower bound and use an approximate lower bound instead. This can be obtained working with an approximation of the log-likelihood function like a first or second order Taylor expansion (in the canonical parameters). A second order Taylor expansion is not only more accurate but also yields a 'working Gaussian distribution' of 'working observations' well known in statistics to tackle inference in GLMs. The original observations are 'lifted' to 'working observations' by a first order Taylor approximation of the (canonical) link function which maps the means to the canonical parameters. The working Gaussian distribution is compatible with (conjugate to) Gaussian priors on eigenvectors and Gaussian distributions of latent factors. This is the approach pursued in this paper where the parameter functions are discretized but additional assumptions of smoothness are made. In particular, in this way (an approximate) Bayesian FPCA for log-intensity functions observed with sparse Poisson counts can be carried out.

Although, given the experience with Gaussian data, the use of an approximate likelihood function based on working observations looks conceptually and computationally straightforward, specific difficulties arise in more general exponential families which require additional efforts. In particular, the point of expansion of the Taylor approximation has to be chosen carefully to make the approximate likelihood work in posterior inference and in the choice of model parameters, and another level of iteration—over points of expansion—is introduced in the algorithm. Furthermore, binary and count data are less informative than data on a continuous scale, and restrictive features of the underlying generative model become more dominant. Correspondingly, the results of analyses with naively selected examples can be disappointing, and the main aim of the paper is to investigate and illustrate the conditions under which Bayesian FPCA for binary and count data using the approximation of the log-likelihood function with working observations does work effectively.

Several purposely constructed examples are presented to enhance an understanding of the features of the generative model for functions to be estimated with binary or count data.

(i) The functions are modeled as interpolation splines at a fixed interpolation design d with N points, but may be observed at points not in d. Thus the number of observation points may be smaller or greater than N, and a first practical question is how to choose the interpolation design d (parsimony of the generative model).
(ii) In particular, there is a trade-off between the degree of smoothness of the functions (the rougher the functions, the more interpolation points are needed) and the stability and accuracy of estimates (the rougher the functions, the more unstable the parameter estimates are).
(iii) There is another trade-off between the similarity of functions (estimable with few eigenfunctions, small sample sizes) and their identifiability (requiring large sample sizes).
(iv) It is just the generative model that conceptually allows for sparse or partial designs from which curves cannot be reconstructed individually. In practice, however, the relative loss of information in binary or count data as compared to Gaussian data, enhances the uncertainties due to sparse designs. In order to compensate by borrowing strength from similar curves (and an assumption of smoothness) many curves are needed.

For comparison, the method proposed by Hall et al. (2008) is applied to most data sets as well. It is implemented in a MATLAB software called PACE (available at http://anson.ucdavis.edu/~btliu/PACE).

The paper is organized as follows. In the next Sect. 2, technical specifications of the proposed method are given. In this section modifications of the variational algorithm for non-Gaussian data are introduced and discussed. In Sect. 3, two examples with binary data are presented, illustrating the interactions between the choice of the interpolation design, smoothness, respectively identifiability, of the functions and sample sizes. In Sect. 4, two examples with Poisson count data are analyzed. The first one simulates a typical application to sparse longitudinal data; the second one is an application to real data. A brief discussion in Sect. 5 concludes.

2 Model and algorithm

The model and algorithm suggested in this article extend the approach described for Gaussian data by van der Linde (2008) to binary and count data. The technical specification given in this section therefore focuses on the modifications necessary to obtain the extension and is brief with respect to general motivations outlined in the previous paper.

2.1 Generative model for binary and count data

We assume that for each of M individuals binary observations or counts are recorded over time yielding N_m-vectors $y_m = (y_{1m}, \ldots, y_{N_m m})^T$ of observations according

to a design $d_m = \{t_{1m}, \ldots, t_{N_m m}\}$ of time points, $m = 1, \ldots, M$. Such observations realize random variables Y_{nm} from a one-parameter exponential family with densities

$$p(y_{nm}|\xi_{nm}) = a(y_{nm}) \exp\big(y_{nm}\xi_{nm} - b(\xi_{nm})\big), \tag{1}$$

with $E(Y_{nm}) = b'(\xi_{nm})$, $\mathrm{Var}(Y_{nm}) = b''(\xi_{nm})$. Observations of different individuals are assumed to be independent, and observations of the same individual at different points in time are assumed to be conditionally independent given the parameters. The means $\mu_{nm} = E(Y_{nm})$ are mapped to the canonical parameters ξ_{nm} by the canonical link function g, $g(\mu_{nm}) = \xi_{nm}$, with $g'(\mu_{nm}) = b''(\xi_{nm})^{-1}$. For binary data the canonical link is the logit link, $g(\mu_{nm}) = \log(\mu_{nm}/(1 - \mu_{nm})) = \mathrm{logit}(\mu_{nm})$, for Poisson counts it is the log-link, $g(\mu_{nm}) = \log(\mu_{nm})$.

The canonical parameters ξ_{nm} are modeled as values of a curve f_m on $\mathbb{R}$ which characterizes the mth individual, that is, $\xi_{nm} = f_m(t_{nm})$. The functions f_m are represented in a space of interpolation splines with respect to a common interpolation design d with N points, $H_{I(d)}$, say. (For further details on $H_{I(d)}$, see Appendix A.1.) The N_m-vector of function values at d_m, f_{md_m}, is related to the N-vector of function values at d, f_{md}, by an interpolation matrix $IP_m : f_{md_m} = IP_m f_{md}$. The interpolation splines $h_{I(f_{md})}$ corresponding to the vectors of function values f_{md} thus form a bundle of curves for which a functional PCA is sought. More precisely, a decomposition of any function $h_{I(f_{md})}$ into a mean function $h_{I(c_d)}$, say, and a residual function $h_{I(f_{md})} - h_{I(c_d)}$ is required, and the space of residual functions is to be spanned by K eigenfunctions $h_{I(a_{kd})}$, say, $k = 1, \ldots, K$, which represent modes of variation within the bundle of curves. Thus with $A = (a_{1d}, \ldots, a_{Kd}) \in \mathbb{R}^{N \times K}$ and s_m denoting a K-vector of unknown coefficients, a generative model for the vector of canonical parameters

$$f_{md} = c_d + A s_m, \quad m = 1, \ldots, M \tag{2}$$

is specified. Thus K is the number of principal components used to reconstruct the vectors of function values and—by interpolation—the functions. Furthermore, in order to enhance smoothness of the mean function and the eigenfunctions, c_d and a_{kd} are represented in a special basis ($N \times r_Q$-matrix) Q_d and ($N \times r_R$-matrix) R_d, respectively, where r_Q and r_R act as smoothing parameters with small values corresponding to smooth functions. Hence $c_d = Q_d \delta$, $a_{kd} = R_d \gamma_k$, and with $G = (\gamma_1, \ldots, \gamma_K)$ the generative model eventually states

$$f_{md} = Q_d \delta + R_d G s_m, \quad m = 1, \ldots, M. \tag{3}$$

(Technical details about the matrices Q_d and R_d are given in Appendix A.1.) Intuitively, Q_d and R_d correspond to eigenfunctions of the underlying latent process, the paths of which are interpolation splines. Because of the interpolation step, the covariance function of the process can be reduced to a covariance matrix, and Q_d and R_d result from a PCA of that matrix. Similarly to the Demmler–Reinsch-basis for splines, the columns of Q_d and R_d represent different frequencies (van der Linde 2003) such that the larger r_Q (respectively, r_R) the more frequencies are incorporated and the more accurately rough functions (interpolation splines) can be approximated. K, r_Q and r_R are model parameters to be selected depending on the data and the distributional assumptions hold conditional on these values.

In principle, different basis functions like those of the Fourier basis could be used to represent vectors of function values f_{md} at d. However, for model choice it is important that the basis functions can be ordered so that the complexity of the functions to be represented is driven by only one parameter, the number of basis functions included. Hence basis functions like B-splines depending on a choice of (variable) knots are not suitable in this approach. Furthermore, the potential of a basis to approximate arbitrary (smooth) functions, which is reflected in the size of the (ordered) basis, refers to a metric in function space. The accuracy of approximation by interpolation splines depends on the accuracy of approximation of the vectors of function values to be interpolated. The matrices Q_d and R_d are optimized by PCA for this approximation in Euclidean space in contrast to evaluations at d of orthogonal basis functions in a Hilbert function space. Thus the key point in the proposed approach is the interpolation step which is compatible but less coherent with different choices of Q_d and R_d.

The model is completed specifying a prior density:

$$\begin{aligned} \delta &\sim N(0, \beta I_{r_Q}), \\ \gamma_k|\sigma_k^2 &\sim N\big(0, \sigma_k^2 I_{r_R}\big) \quad \text{independently for } k = 1, \ldots, K, \\ \sigma_k^2|\alpha_{A0}, \beta_{A0} &\sim IG(\alpha_{A0}, \beta_{A0}) \quad \text{independently,} \end{aligned}$$

and

$$s_m \sim N(0, \beta I_K).$$

Here IG denotes an inverse Gamma distribution, and $\alpha_{A0} = \beta_{A0} = \beta^{-1} = 10^{-3}$ are chosen. With $s = (s_1, \ldots, s_m)$ and $\sigma_A = (\sigma_1^2, \ldots, \sigma_K^2)$ the joint prior density is given by

$$p(s, \delta, G, \sigma_A) = \prod_{m=1}^{M} p(s_m)\, p(\delta) \prod_{k=1}^{K} p\big(\gamma_k|\sigma_k^2\big) \prod_{k=1}^{K} p\big(\sigma_k^2|\alpha_{A0}, \beta_{A0}\big).$$

2.2 The variational algorithm: general issues

In general, in order to derive an approximate posterior density q of unknown quantities $z = (z_1, \ldots, z_J)$ by applying a variational algorithm, a lower bound L_q of the log-marginal density of the data y_{all},

$$\log p(y_{\text{all}}) \geq L_q = E_q\big(\log p(y_{\text{all}}, z)/q(z)\big),$$

is to be maximized. If q is factorized as $q(z) = \prod_{j=1}^{J} q_j(z_j)$ this is achieved taking iteratively

$$q_j(z_j) \propto \exp\big(E_{q\setminus j}\big(\log p(y_{\text{all}}, z)\big)\big), \tag{4}$$

where $q\setminus j$ denotes the current joint density of the z_i without z_j.

The evaluation of the expectation $E_{q\setminus j}(\log p(y_{\text{all}}, z))$ can be a problem, particularly if the likelihood and the prior are not conjugate as in applications with discrete data and Gaussian priors. Several strategies were suggested to make the algorithm tractable (in closed form) introducing further approximations:

(1) Problem specific lower bounds $\widetilde{L}_q \leq L_q$, for binary data (Jaakkola and Jordan 1997; Tipping 1999).
(2) Approximate lower bounds $\widetilde{L}_q \approx L_q$, induced by approximations of $E_{q\setminus j}(\log p(y_{\text{all}}, z))$, respectively $E_{q\setminus j}(\log p(y_{\text{all}}|z))$ (Braun and McAuliffe 2008).

Especially for Poisson data, no useful lower bound could be derived yet. But, referring to standard statistical practice, an approximate lower bound can be suggested (for $z = (s, \delta, G, \sigma_A)$ and $y_{\text{all}} = (y_{1d_1}, \ldots, y_{Md_M})$). In GLMs, for one-parameter exponential families second order Taylor expansions with updated points of expansion of the log-likelihood function are commonly used to compute ML-estimates or posterior modes (McCullagh and Nelder 1983; Green and Silverman 1994). A second order Taylor expansion of the log-likelihood function is equivalent to a *working Gaussian distribution* of *working observations* which then is conjugate to Gaussian priors and allows for updates of the approximate posterior distribution in closed form. Jaakkola and Jordan (1997) already briefly considered a second order Taylor expansion (*Laplace approximation*) in the binary case but they used only the prior mode as point of expansion yielding—not surprisingly—comparatively bad results.

The marginal log-density of the data $\log p(y_{\text{all}})$, respectively the lower bound L_q, is also used for model selection: Essentially, the maxima obtained for fixed model parameters K, r_Q and r_R are compared, and the parameters giving the highest maximum are chosen. For non-Gaussian errors though, the choice of the model parameters also depends on the choice of the point of expansion in the Taylor approximation of the log-likelihood, and hence the optimization of K, r_Q and r_R has to be iterated as well. An algorithm for parameter estimation with alternating choices of the point of expansion and K, r_Q and r_R is detailed in Sect. 2.4.

2.3 Working observations

Write $l(\xi|y)$ for the log-likelihood with a generic single observation y and a generic canonical parameter $\xi \in \mathbb{R}$, and denote by $\xi_0 \in \mathbb{R}$ a point of expansion. For the second order Taylor expansion $\widehat{l}(\xi|y)$ in ξ_0, one has $l(\xi|y) \approx \widehat{l}(\xi|y) = l(\xi_0|y) + l'(\xi_0|y) \times (\xi - \xi_0) + \frac{1}{2} l''(\xi_0|y)(\xi - \xi_0)^2$ with $l'(\xi_0|y) = y - b'(\xi_0)$ and $l''(\xi_0|y) = -b''(\xi_0)$. Define the working observation $w(y|\xi_0)$ related to the original observation y and depending on the point of expansion ξ_0 by

$$w(y|\xi_0) = \xi_0 - l'(\xi_0|y)/l''(\xi_0|y) = \xi_0 + b''(\xi_0)^{-1}\big(y - b'(\xi_0)\big). \tag{5}$$

Intuitively, a working observation is a first order Taylor approximation (in $\mu_0 = g^{-1}(\xi_0)$) of the link-transformed original observation:

$$\begin{aligned} & y \to g(y) \\ & \approx\ g(\mu_0) + g'(\mu_0)(y - \mu_0) = \xi_0 + b''(\xi_0)^{-1}\big(y - b'(\xi_0)\big). \end{aligned}$$

Note that in the case of binary data strictly the (logit) link function is not defined for values 0 and 1, but the approximation is defined for $\mu_0 \notin \{0, 1\}$. For the working observation, we have

$$\widehat{l}(\xi|y) = -\frac{1}{2}\big(w(y|\xi_0) - \xi\big)\big(-l''(\xi_0|y)\big)\big(w(y|\xi_0) - \xi\big) + l(\xi_0|y) - \frac{1}{2}\frac{(l'(\xi_0|y))^2}{l''(\xi_0|y)}. \tag{6}$$

The quadratic form is used as working Gaussian log-density with known variance, and the remainder does not depend on ξ.

Now, recalling that in our functional model $\xi_{nm} = f_m(t_{nm})$ and $f_{md_m} = IP_m f_{md}$, points of expansion $f^0_{md_m} = IP_m f^0_{md}$ can be summarized in N-vectors f^0_{md} (inducing an interpolation spline) for each of the M individuals. The contribution of the mth individual to the likelihood then can be treated as if independently

$$w_m\big(f^0_{md}\big) \sim N\big(IP_m f_{md}, D\big(f^0_{md}\big)\big),$$

where in obvious notation w_m is the $\dot{N}_m$-vector of working observations obtained from y_m and $D(f^0_{md})$ is the diagonal matrix with entries $b''(f^0_{nm})^{-1}$, $n = 1, \ldots, N_m$. Hence the variational algorithm derived for Gaussian observations $y_m \sim N(IP_m f_{md}, \sigma^2 I_{N_m})$, described in Appendix A.2, can be adapted as outlined in Appendix A.3 and used as a building block when iterating over points of expansion.

2.4 The specific variational algorithm

The following procedure is suggested. Let $F = (f_{1d}, \ldots, f_{Md})$ be the matrix of all unknown function values at the interpolation design d.

1. Choose a first matrix of points of expansions $F^0(1)$, specifying initial values of F. Estimate the degree of smoothness of the mean function $h_{I(c_d)}$ by $r_Q(1)$ (recalling $c_d = Q_d\delta$ where $Q_d \in \mathbb{R}^{N\times r_Q}$).
2. Set $K(1) = 2$, and $r_R(1) = r_Q(1) + 2$.
   Run the variational algorithm with $K = K(1)$, $r_Q = r_Q(1)$, $r_R = r_R(1)$ and working observations based on $F^0(1)$.
   Build a new matrix $F^0(2)$ from the resulting posterior means of $f_{md}, m = 1, \ldots, M$.
3. Determine new model parameters $K(2)$, $r_Q(2)$ and $r_R(2)$ maximizing the approximate lower bound obtained when running the variational algorithm with working observations based on $F^0(2)$. The maximum is attained at corresponding posterior means of f_{md} defining a new matrix $F^0(3)$.
4. Repeat step 3 until the model parameters do not change any more. Keep the approximate posterior distribution obtained in the previous step as the final result.

In detail, more specifications are needed.

Step 1. Choosing a good initial matrix of points of expansion is important in order to scale the model correctly and to provide a reasonably accurate approximation of the

log-likelihood function and hence the lower bound. One may start with an estimate of the mean function, respectively an estimate c_d^0 obtained using all data points. Any smoothing procedure could be used to obtain such a fit. Here (in order to implicitly validate the programming) a local linear smoothing procedure as described by Hall et al. (2008, their Appendix A.1) was applied. The required smoothing parameter can be determined by cross-validation. For all examples in this article, robust cross-validation aiming at the stability of the estimate when leaving out vectors y_m in turn was used. The fit of the estimated mean function, respectively of c_d^0, in the span of r_Q columns of Q_d yields $r_Q(1)$. The fit was required to be highly accurate in order not to undersmooth the eigenfunctions initially (setting $r_R(1) = r_Q(1) + 2$). The matrix $F^0(1)$ was obtained with random modifications of the mean function like $f_{md}^0(1) = 0.75 \cdot 1_N + 0.5 \cdot u_m \cdot c_d^0$, where u_m is a realization of a random variable distributed uniformly on [0, 1].

Step 2. $K(1) = 2$ corresponds to two eigenfunctions, a standard default value in PCA. As the residual functions tend to be less smooth than the mean function, $r_R(1) > r_Q(1)$ is a reasonable first choice.

Step 3. Choosing the model parameters based on an *approximate* lower bound of the marginal density is substantially different from the variational algorithm with exact lower bound (for instance, with genuine Gaussian observations) and not guaranteed to give the best values. In order to assess this step, the values of the approximate lower bound can be related to the goodness of fit of the estimated (known) functions in examples with simulated data. This is exemplified in Sect. 4.1 for Poisson data. The range for the search of model parameters was initially set to $K(2) \in \{1, 2, 3, 4\}$, $r_Q(2) \in \{r_Q(1) - 2, r_Q(1) - 1, r_Q(1), r_Q(1) + 1, r_Q(1) + 2\} =: \{r_Q(1) \pm 2\}$ and $r_R(2) \in \{3, \ldots, 10\}$. Subsequently neighborhoods of the model parameters of the type ± 2 were searched, and whenever a marginal value was selected in the maximization the search was extended by two more values in that direction. The searches are the computationally intensive part of the method and can, of course, be modified if that is felt to be necessary.

In general, the algorithm was kept as simple and parsimonious as possible, not only because of the computational costs but also because the estimation of high dimensional parameters with sometimes small sample sizes tends to be unstable. Therefore, for example, in the last step the second to last estimate instead of the last estimate is retained, and in the alternating estimation of the model parameters K, r_Q, r_R, and of the function values F only one run of the variational algorithm is used. Updating the point of expansion several times for fixed K, r_Q and r_R can enhance but also deteriorate the performance.

The variational algorithm was implemented in MATLAB.

2.5 Evaluation of performance

In order to assess the accuracy in recovering the mean function and the canonical parameter functions, the discretization at the interpolation design d is evaluated and relative measures of fit

$$rmean = 1 - \frac{\|c_d - \widehat{c}_d\|^2}{\|c_d\|^2},$$

$$rcan = 1 - \frac{1}{M}\sum_{m=1}^{M}\frac{\|f_{md} - \widehat{f_{md}}\|^2}{\|f_{md}\|^2}$$

are reported.

3 Examples with binary data

As a special case of a one-parameter exponential family (1), consider densities from Bernoulli distributions $B(1, \mu_{nm})$, $n = 1, \ldots, N_m$, $m = 1, \ldots, M$, where $a(y_{nm}) = 1$ and $b(\xi_{nm}) = \log(1 + e^{\xi_{nm}})$. Interest is in the canonical parameter functions with values $f(t_{nm}) = \xi_{nm} = logit(\mu_{nm})$, referred to for brevity as logit functions. The inverse logit transform yields the related probability functions. The observation of the mth logit function is given by an N_m-vector y_m with binary values. Few binary data points are not very informative about the smoothness of a logit function. The main topic of this section therefore is the interaction between the smoothness of the logit functions and the sample sizes needed to reconstruct them.

3.1 Very smooth similar curves

A simple set of $M = 40$ similar increasing probability functions is considered, 20 of which together with the mean function are displayed in Fig. 1 (top left). The 40 logit curves f_m,

$$f_m(t) = 0.8\big(-2 + \alpha_m(1.6t + \delta_m)^2\big), \quad t \in [0, 1],$$

are shown in the top right panel of Fig. 1. The variation is due to a random choice of $\alpha_m = 1 - 0.3u_{1m}$ and $\delta_m = 0.8u_{2m}$, where u_{1m}, u_{2m} are realizations of random variables uniformly distributed on [0, 1]. Based on an equidistant grid with 30 points in [0, 1], a conventional PCA of the vectors of function values suggests essentially one mode of variation (with explained variance of 0.995), which can be interpreted as the spread within the set of curves (Fig. 1, bottom right) induced by different slopes of the probability functions. The eigenfunction is simply a straight line (Fig. 1, center right). In order to illustrate how the curves are reflected in binary data, the two most extreme curves f_{35} (with $\alpha_{35} = 0.9968$, $\delta_{35} = 0.7513$) and f_{37} (with $\alpha_{37} = 0.8375$, $\delta_{37} = 0.0474$) are displayed in Fig. 1 (left column).

3.1.1 Analyses with small sample size

Although the curves are simple and do differ by the degree of increase, they are not easily discriminated with binary data. We first show that a sample size of $N_m = 10$ observations per curve is insufficient to recover them.

Interpolation splines with few points As the curves are very smooth, only few points are needed to fit them with interpolation splines. Hence an interpolation design d might be chosen as a regular grid over [0, 1] with only $N = 10$ points. The mean

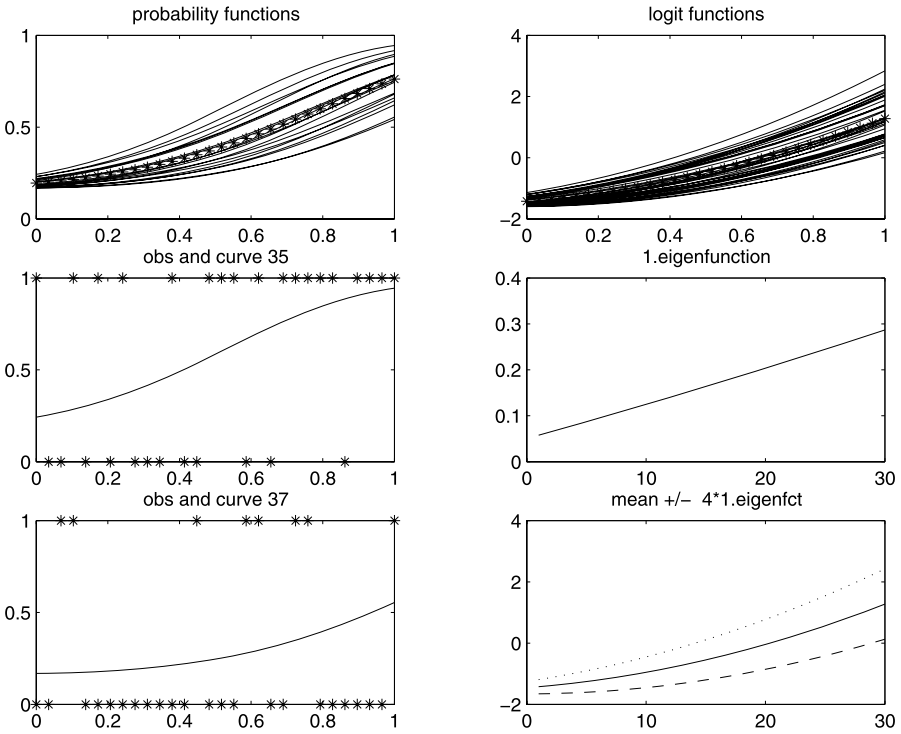

Fig. 1 *Top left*: 20 probability functions with mean function (of 40, *stars*). *Center* and *bottom left*: two probability curves with $N_m = 30$ data points (*stars*). *Top right*: 40 canonical parameter functions with mean function (*stars*). *Center right*: first eigenfunction. *Bottom right*: logit mean function (*solid line*) plus (*dotted line*) and minus (*dashed line*) four times the first eigenfunction

Table 1 Example 1, results with $N = N_m = 10$

Run	K	r_Q	r_R	*rmean*	*rcan*
1	2	6	8	0.8136	0.6877
2	1	4	3	0.9386	0.8062
3	1	3	3	0.9871	0.8407

function is initially fitted with $r_Q(1) = 6$ basis functions. The application of the algorithm then yields the values listed in Table 1, where the measures of fit are based on evaluations at d.

Clearly, the iterations over the points of expansion improve the estimates, and the one mode of variation is correctly identified. However, the individual functions cannot be fully separated as shown in Fig. 2, panel $(1, 2)$.

The fit obtained with PACE (the method proposed by Hall et al. 2008) is shown in panel $(1, 3)$ of Fig. 2. (PACE was run using the options 'kernel = gauss', '*selection_k* = BIC1', '*maxk* = 10', '*regular* = 2', and the defaults otherwise.) With PACE, three principal components were suggested, resulting in *rmean* = 0.9497 and *rcan* = 0.6573, an overall performance worse than the one obtained with the genera-

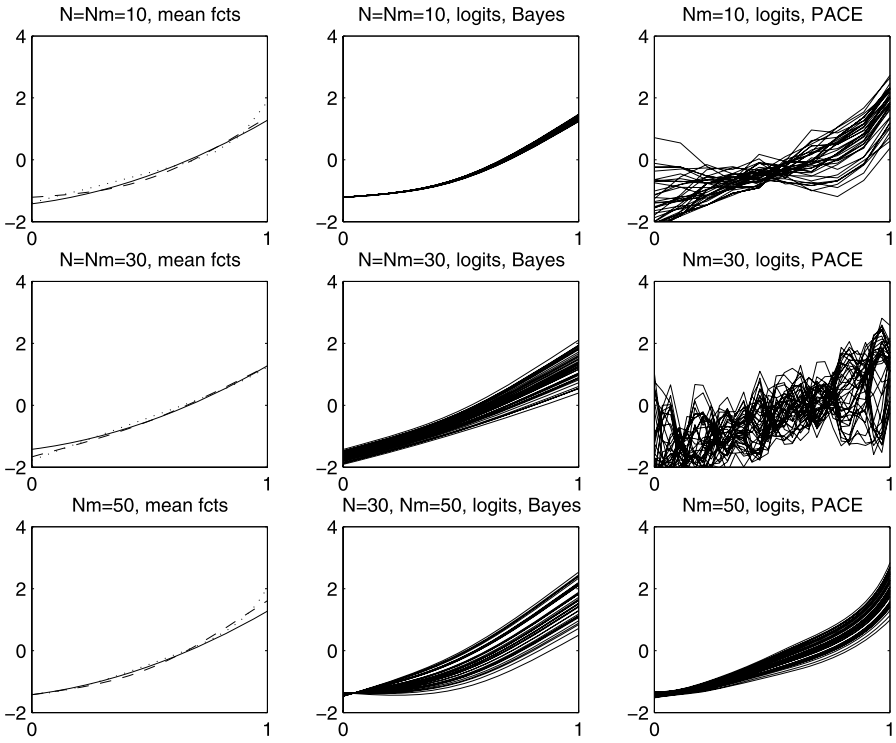

Fig. 2 *First column*: true (*solid line*) and estimated mean function (Bayes: *dashed line*, PACE: *dotted line*). *Second* and *third column*: estimated canonical parameter functions (Bayes: *2 column*, PACE: *3 column*) with estimates based on $N_m = 10$ (*top*), $N_m = 30$ (*center*), $N_m = 50$ (*bottom*) observations

tive model. PACE does discriminate curves based on such sparse data but incorrectly, and exhibits problems in smoothing adequately.

Interpolation splines with more points Without prior knowledge about the smoothness of the curves, a design with $N = 10$ interpolation points would be regarded as too sparse. A more appropriate uninformed all purpose design would be chosen as one with 30–50 points, and we next try an equidistant grid with $N = 30$ points. Thus the number of unknown parameters (points of expansion) is increased, the matrix F of unknown function values being $N \times M$, although in the parameterization $f_{md} = Q_d\delta + R_d G s_m$ the dimensionality r_Q of δ, $r_R \times K$ of G, and $K \times M$ of s_m do not depend on N. We use the same sample size $N_m = 10$ as before so that the setup corresponds to an applied study like the one described by Hall et al. (2008) (with 42 patients and binary observations between 3 and 12 points in time each). The initial fit of the mean function at d suggests $r_Q(1) = 7$. The results obtained are summarized in Table 2 where the measures of fit are based on evaluations at $N = 30$ points.

For comparison also *rmean* and *rcan* of the previous analysis with $N = 10$ were recalculated based on an evaluation of the finally estimated interpolation spline at d with now $N = 30$ points yielding $rmean = 0.9887$, $rcan = 0.8307$. With $N = 30$, the

Table 2 Example 1, results with $N = 30$, $N_m = 10$

Run	K	r_Q	r_R	*rmean*	*rcan*
1	2	7	9	0.8553	0.7123
2	1	5	3	0.8760	0.7302
3	1	3	3	0.9912	0.8272

Table 3 Example 1, results with $N = N_m = 30$

Run	K	r_Q	r_R	*rmean*	*rcan*
1	1	4	3	0.9922	0.9192
2	1	3	3	0.9907	0.9182
PACE	9			0.9834	0.4663

mean function is fitted well, slightly better than before, but the fit of the canonical parameter functions is poor, worse than in the model with 10 interpolation points. The individual functions can hardly be discriminated from the mean function. The iteration only achieves sufficient smoothing of the mean function and of the eigenfunction, thus recovering again correctly the one mode of variation.

3.1.2 Analysis with increasing sample sizes

With a standard interpolation design of $N = 30$ points, $N_m = 30$ binary observations per curve at a regular grid still yield unsatisfying fits of the individual curves (as evident in Fig. 2, panel (2, 2)). PACE results in a severe lack of fit.

The Bayesian results are better than those obtained with PACE, which does not succeed in smoothing appropriately so that the fit of the individual curves is worse than that with fewer observations as shown in Table 3. Figure 2 (last row) shows that at least $N_m = 50$ binary observations of each curve are needed to reconstruct the spread within the set of functions. With $N_m = 50$, PACE identifies the one dominating mode of variation, but the fit for the mean function, *rmean*(PACE) $= 0.9676$, as well as that for the canonical parameter functions, *rcan*(PACE) $= 0.8966$, is worse than that obtained in the Bayesian approach with *rmean*(BAYES) $= 0.9767$, *rcan*(BAYES) $= 0.9221$.

3.2 Curves differing by the degree of smoothness

As a second example, we consider a set of $M = 50$ curves which consists of essentially Gaussian densities with zero mean and varying variances. They are truncated and shifted in order to construct logit functions—shown in Fig. 3 (top right)—corresponding to probability functions with values between 0.1 and 0.9. More precisely, starting with g_m as $N(0, \sigma_m^2)$-density, where σ_m^2 is taken from a regular grid on [0.2,0.7] with $M = 50$ points, $f_m(t) = 0.4(g_m(t) - 5)$, $t \in [-2, 2]$. The unimodal functions differ by the height of the peak at zero and by curvature. Essentially, the shape is reflected with binary data as exemplified in the top left panel of Fig. 3.

The interpolation design d was chosen to be an equidistant grid over $[-2, 2]$ with $N = 50$ points which is sufficient to recover the true functions by interpolation. The

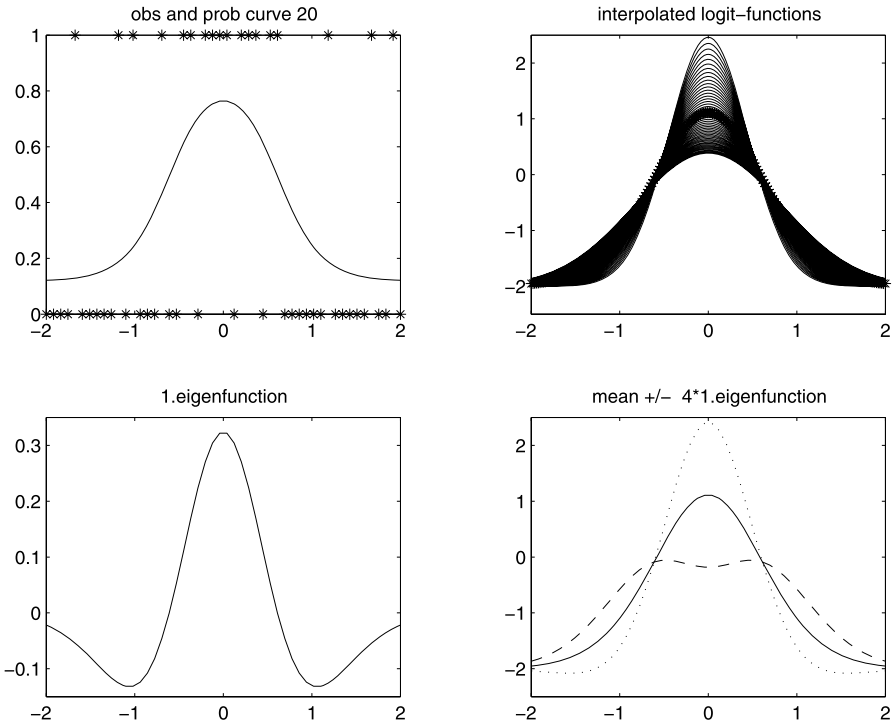

Fig. 3 *Top left*: A probability curve with 50 data points. *Top right*: 50 logit functions smoothly interpolated at 50 equidistant points with mean function (*stars*). *Bottom left*: first eigenfunction resulting from PCA with vectors of true function values. *Bottom right*: logit mean function (*solid line*) plus (*dotted line*) and minus (*dashed line*) four times the first eigenfunction

mean curve can be fitted with 7 basis functions, the steepest logit curve requires 9 basis functions, and the residual curves between 9 and 11 basis functions. Because of the more pronounced structural difference, the individual curves can be expected to be more easily identifiable, but the determination of the number r_R of basis functions to reconstruct the residual functions may become a problem. In a conventional PCA with $M = 50$ vectors of true function values, the main mode of variation (corresponding to an explained variance of 0.984) is easily detected (see Fig. 3, bottom row).

3.2.1 The performance of the algorithm

The performance of the algorithm is compared for increasing sample sizes N_m with d_m chosen as equidistant grids over $[-2, 2]$. Model parameters and measures of fit of the estimates from final runs of the algorithm with different sample sizes are summarized in Table 4; *rmean* and *rcan* are based on the evaluation at 50 points.

50 binary observations per curve prove to be insufficient: Only the mean function is reasonably recovered, the first eigenfunction is undersmoothed, and all individual functions are estimated by the mean function, that is, not at all discriminated. With

Table 4 Example 2, results with increasing sample sizes

N_m	Runs	K	r_Q	r_R	*rmean*	*rcan*
50	2	1	5	3	0.9919	0.9537
100	4	1	5	5	0.9943	0.9695
200	2	1	7	7	0.9983	0.9822
50	PACE	2			0.9866	0.9563
100	PACE	3			0.9844	0.9563

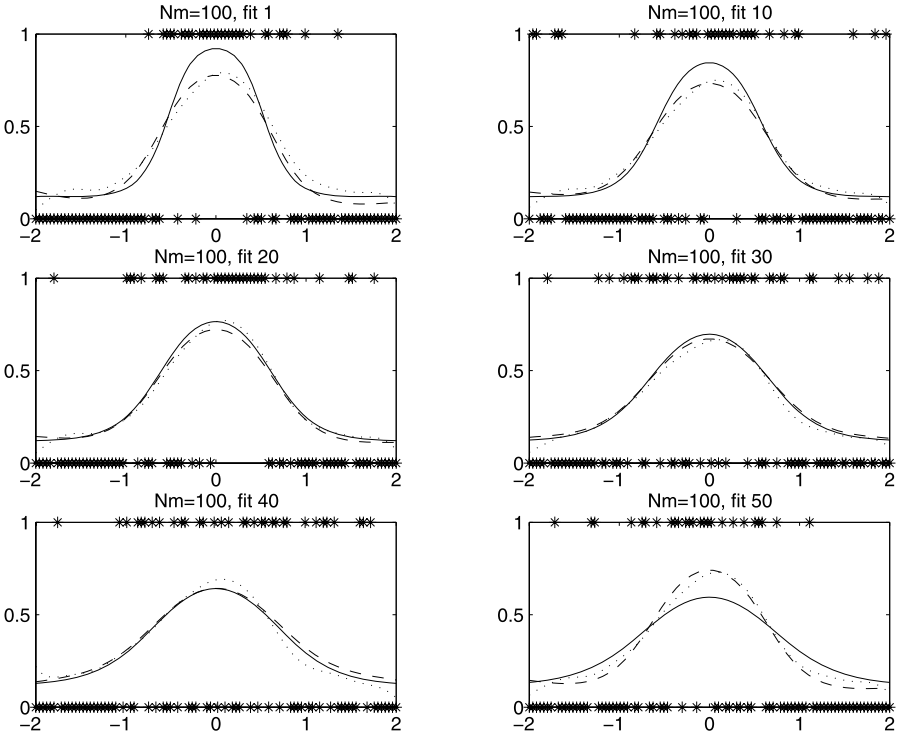

Fig. 4 Six selected probability functions (*solid lines*) and their Bayes-estimates (*dashed lines*), respectively PACE-estimates (*dotted lines*), along with the 100 observed data points for each curve

100 observations per curve, the estimate of the mean function is again acceptable; the shape of the first eigenfunction becomes visible, although it is still undersmoothed. Consequently, the principal mode of variation is recovered with the fitted curves, but the smoothest curves are undersmoothed, while the roughest curves are oversmoothed. Thus in the middle range the fits are adequate, whereas the extreme curves are missed. With 200 binary observations of each curve, the fits improve, yet the most extreme curves are not fully recovered.

In Fig. 4, the estimates based on $N_m = 100$ observations of six selected logit curves of increasing smoothness are displayed, showing that the most extreme curves are not perfectly reconstructed. The oversmoothing (undersmoothing) of the rough-

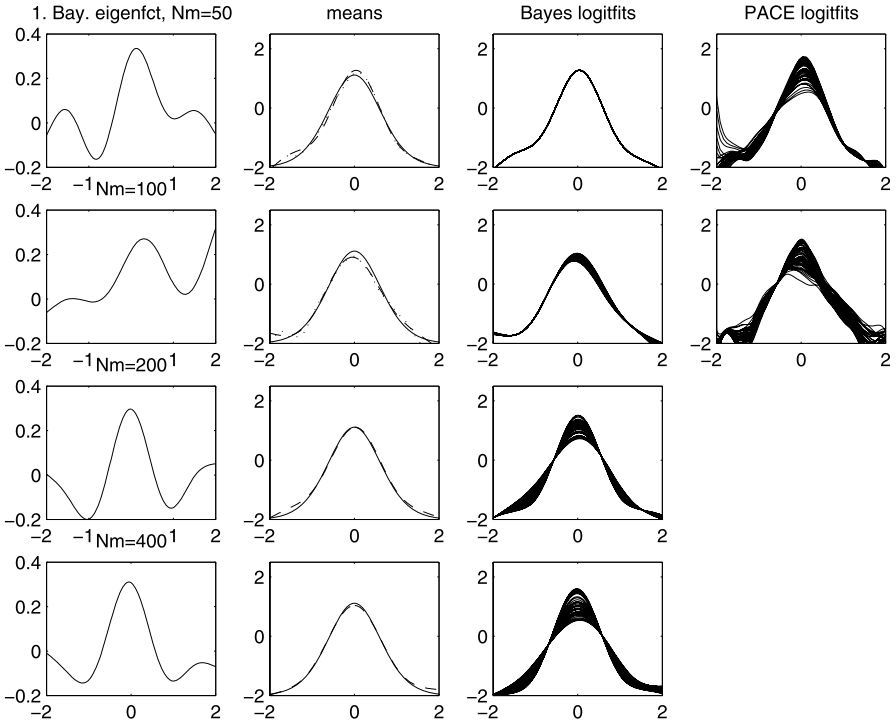

Fig. 5 *First column*: Bayes estimate of (first) eigenfunction. *Second column*: true (*solid line*) and estimated (Bayes: *dashed line*, PACE: *dotted line*) mean function. *Third column*: estimated canonical parameter functions, all based on Bayesian FPCA. *Fourth column*: canonical parameter functions estimated with PACE. Rows correspond to increasing sample sizes. All Bayesian estimates based on one run use $N = 50$, the initial values $K = 1$, $r_Q = r_R = 7$ and the same point of expansion

est (smoothest) curves illustrates that the generative model is set up as a multivariate multiple regression model with emphasis on common features of all functions. Running PACE for this set of curves (with options as described in Sect. 3.1.2) yields measures of fit which are slightly worse (see Table 4), but the PACE-fitted curves are considerably undersmoothed with more eigenfunctions chosen to reconstruct the curves. For the more extreme curves, the peaks are missed as with the generative model (also shown in Fig. 4). In Fig. 5 (last column), all PACE-fits are shown for $N_m = 50$ and $N_m = 100$.

While the Bayesian analysis never takes more than an hour, the computing time of PACE grows with sample size: several hours for $N_m = 100$, more than 36 hours for $N_m = 200$. This is probably due to the determination of the smoothing parameters by cross-validation in PACE, whereas the choice of K, r_Q and r_R is based on a guided exploration (as described in Sect. 2.4).

3.2.2 The pure effect of sample size

In order to single out the effect of sample sizes N_m, the same initial setup with $K = 1$, $r_Q = r_R = 7$ and the same first point of expansion (based on $N_m = 50$) was chosen.

Table 5 Example 2, results with one run

N_m	*rmean*	*rcan*
50	0.9920	0.9551
100	0.9904	0.9621
200	0.9967	0.9791
400	0.9982	0.9908

Then just one run was carried out with sample sizes $N_m \in \{50, 100, 200, 400\}$. The improvement with increasing sample size, illustrated in Fig. 5, is as described above, and the measures of fit are listed in Table 5.

4 Examples with Poisson count data

Poisson densities are another example of a one-parameter exponential family with $a(y_{nm}) = -\log(y_{nm}!)$ and $b(\xi_{nm}) = e^{\xi_{nm}} = \mu_{nm}$. Hence the canonical parameter functions are log-transformed intensity functions. The observation of the mth intensity function is an N_m-vector y_m of counts. Two examples are presented in this section. The first example simulates an applied study with sparse data. Technically, the focus in this example is on the choice of the model parameters K, r_Q and r_R based on the values of the approximate lower bound, the relation of which to the measure of fit is explored. The second example is an investigation of real data.

4.1 Sparse longitudinal Poisson counts

Imagine that over one year the frequency of activities (doing sports, watching a TV program, having a vegetarian meal or the like) of $M = 50$ persons has been recorded weekly. Assume that week 1 corresponds to the first week of the year in January, and that there are missing data such that the number of records (weeks) range from 1 to 52 with mean 22.62 and standard deviation 14.09. For each of the subjects the number N_m of records was randomly chosen from the set $\{1, 2, \ldots, 52\}$, and subsequently the N_m points were randomly chosen from that set. A set of intensity functions over the domain $[1, 52]$ yielding such data is shown in the first row of Fig. 6. The intensity functions were generated from functions $g_m(t) = 2 + \alpha_m(1 + \cos(v_m(t))$ with random factors $\alpha_m = 1.5 + 2u_{1m}$, where u_{1m} are realizations of a random variable uniformly distributed on $[0, 1]$. Values of $t \in \{15, 16, \ldots, 66\}$ were used to yield 52 function values. The transformations of t were $v_m(t) = 1.025\pi - 2\pi(t \pm shift(m))/79$. Whether the shift was added or subtracted was randomly chosen, and also the shift was random: $shift(m) = \lceil 7u_{2m} \rceil$, the nearest integer greater than or equal to $7u_{2m}$, and the values u_{2m} are realizations of a random variable uniformly distributed on $[0, 1]$. The 50 log-intensity functions together with the log-transformed mean function (based on all functions) are depicted in panel $(2, 1)$ of Fig. 6.

The displays show that over one year the frequency of activities tends to increase in summer, but the curves differ with respect to the individual dynamics. For the generative model, the interpolation design $d = \{1, 2, \ldots, 52\}$ with $N = 52$ points was

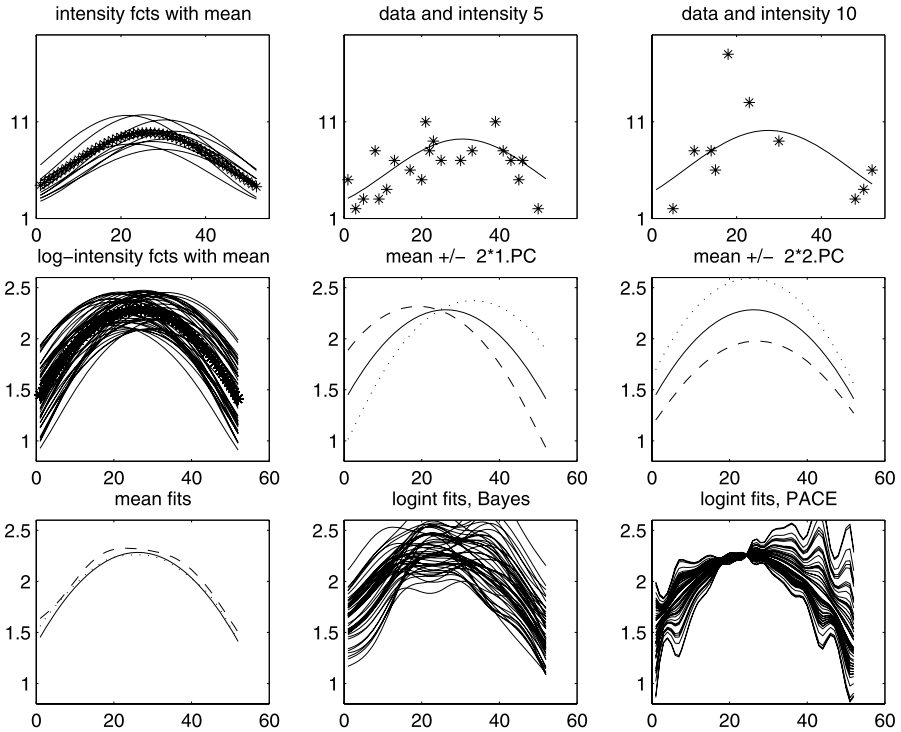

Fig. 6 *Top left*: 10 intensity functions with mean function (based on 50 curves, *stars*). *Top center* and *right*: two intensity curves with observations. *Second row left*: 50 log-intensity functions with mean function (*stars*). *Second row center* and *right*: the log-intensity mean function (*solid line*) plus (*dashed line*) and minus (*dotted line*) two times an eigenfunction. *Bottom left*: true log-intensity mean function (*solid line*) with Bayes-fit (*dashed line*) and PACE-fit (*dotted line*). *Bottom center*: Bayes-fits of 50 log-intensity functions. *Bottom right*: PACE-fits of 50 log-intensity functions

chosen. A conventional PCA of the 50 vectors of true function values at these points reveals two modes of variation (corresponding to a cumulative explained variance of 0.7248 and 0.9997, respectively): (i) the location of peaks, (ii) the level, respectively increase, of activities. The effects of eigenfunctions are shown in Fig. 6 (second row). The initial estimate of the mean function can be fitted with $r_Q(1) = 6$ basis functions. The algorithm summarized in Table 6 takes two runs, essentially with the initial estimates being smoothed in the second run. For comparison, also the measures of performance based on a complete data set (that is, with 52 observations of each curve) are given in Table 6. Clearly, the fit is improved.

The scatter plots in Fig. 7 reveal that there is a positive correlation between the values of the approximate lower bound and the measure of fit *rcan* in the two searches for the model parameters. It is stronger in the second search (confirming $K = 2$, $r_Q = 5$, $r_R = 3$) than in the first search. The example illustrates that initially the choice of model parameters based on the approximate lower bound may not be optimal, but iterating over the points of expansion it can be expected to improve.

Table 6 Example 3, measures of fit

Run	K	r_Q	r_R	*rmean*	*rcan*
1	2	6	8	0.9985	0.9944
2	2	5	3	0.9995	0.9958
Complete	2	5	3	0.9994	0.9986
PACE	1			0.9997	0.9934

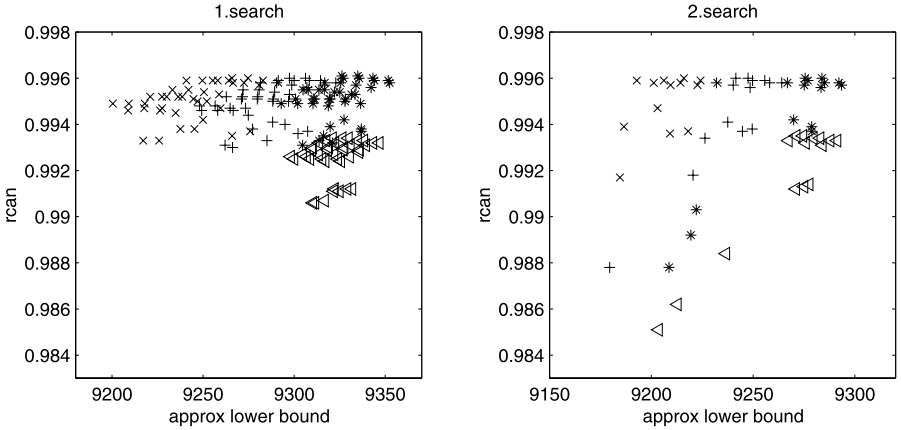

Fig. 7 Scatter plots of the values of the approximate lower bound versus those of the measure of fit *rcan* for the first (*left panel*) and second (*right panel*) search of model parameters K, r_Q, r_R. The symbols indicate the values of K : $\times = 4$, $+ = 3$, $* = 2$, triangle $= 1$

PACE run for the same data set (with option 'regular = 0' and other options as specified before) yields a good fit of the mean function but the one identified eigenfunction (explaining 47.35% of the variance) and the fitted individual curves are undersmoothed. The Bayesian and PACE fits are displayed in Fig. 6 (third row).

Finally, for six selected intensity curves the Bayesian fit along with the 95% credible intervals at $\{1, 2, \ldots, M\}$ are displayed in Fig. 8, and also the PACE fits are shown. The credible sets are based on 1000 simulated values from the approximate joint posterior distribution. Apart from curve 50, the credible sets cover the true intensity functions and the PACE fits.

4.2 Real data

Achcar et al. (2008) describe an experiment with a total of $M = 14$ rats, 8 of which are of the Wistar species and 6 of the War species. The rats were treated with saline and oxytocin, and counts of grooming were recorded every 5 minutes after application of the treatment. Thus after one hour $N_m = 12$ measurements were obtained for each rat. Here only the count data obtained under treatment with oxytocin is used (given in Table 1 of the aforementioned paper). The assumption of a Poisson distribution of the counts is applied, although there is some indication of overdispersion. One question of interest is whether there is a difference between the two species.

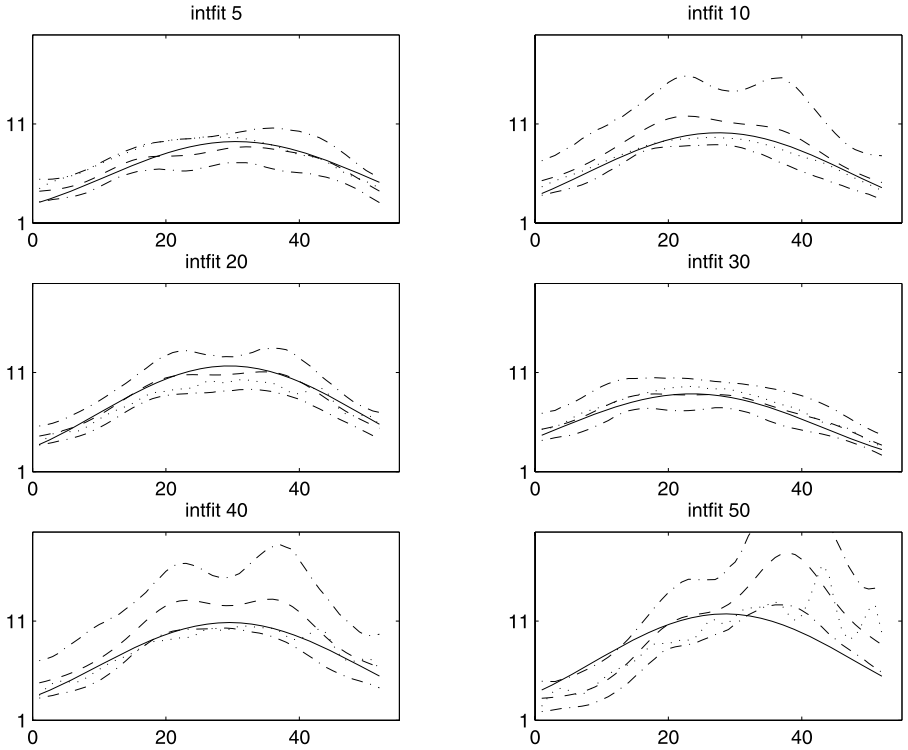

Fig. 8 Six true intensity functions (*solid lines*), their PACE-estimates (*dotted lines*) and their Bayes-estimates (*dashed lines*) together with credible sets (*dashed-dotted lines*) based on sparse Poisson data

To illustrate the data, individual fits of the (log-) intensity functions based on local linear smoothing are shown in Fig. 9 (first row). These fits are rather rough and not suited for a naive PCA based on the 12-dimensional vectors of function values. Running PACE (with default options and '$regular = 2$', '$maxk = 10$') suggests that there is only one mode of variation. It describes the increase of activity (grooming) of the rats up to 35 minutes after treatment and a second minor peak, respectively continuous decrease, ten minutes later (see Fig. 9, second row).

The Bayesian data analysis with $N = N_m = 12$ takes 5 runs and eventually also suggests one mode of variation using $q = 3\,(r = 5)$ basis functions to represent the mean function, respectively the individual residual functions. The results are similar to those obtained with PACE, only the mean function is estimated to be more symmetric about the midpoint of the time interval (30 minutes), and the range of variation in increase after treatment is slightly greater. The interpretation of the one identified eigenfunction is the same (cp. Fig. 9, third column). There is no indication that the two species differ in their reaction curves. In Fig. 10, the curves of the six War rats are displayed along with the observed counts and the fits (local linear smoothing ('naive'), PACE, Bayesian). Furthermore, the 95% credible sets (based on 1000 sim-

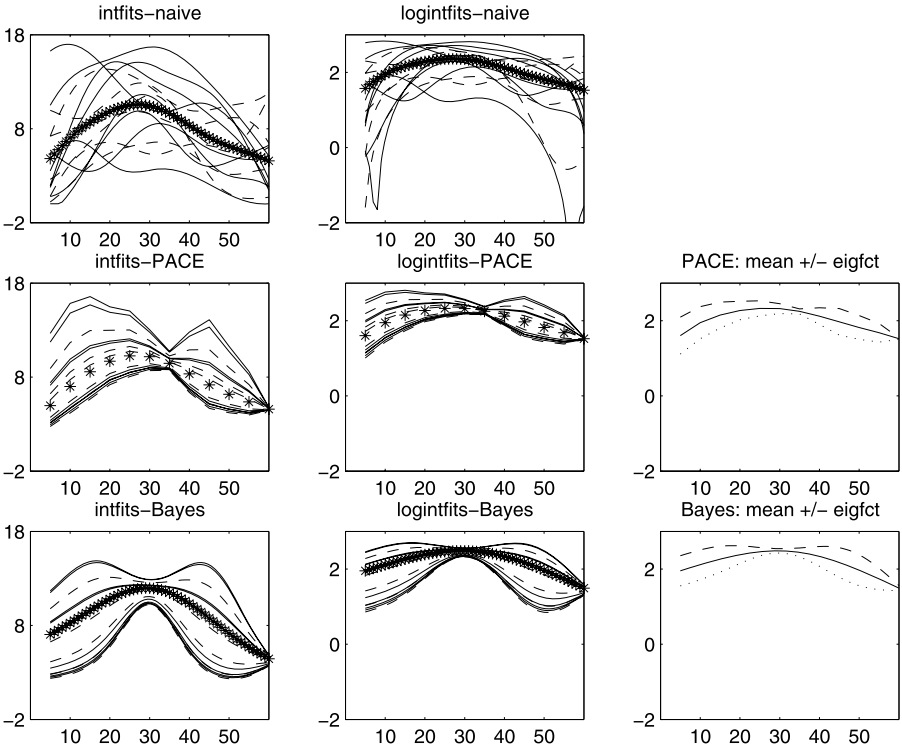

Fig. 9 *First column*: fits of 14 intensity curves based on individual local linear smoothing (*top*: naive), PACE-FPCA (*center*) and Bayes-FPCA (*bottom*). *Solid lines* indicate rats of the Wistar species, *dashed lines* rats of the War species. *Second column*: the corresponding fits of the log-intensity functions. *Third column*: estimated log-intensity mean function (*solid line*) plus (*dotted line*) and minus (*dashed line*) the first eigenfunction. Estimates are based on PACE (*center*) and Bayes (*bottom*)

ulated values from the posterior distribution) for the 12 function values are shown (linearly interpolated).

The fits obtained with PACE lie—apart from very few function values—well within the credible sets, while the naive individual fits often are not covered.

5 Discussion

5.1 Summary of experiences

It was demonstrated by van der Linde (2008) that PCA based on a generative model and a variational algorithm can be extended to functional data if a Demmler–Reinsch(-like) basis of interpolation splines is used where the interpolation design is an equidistant grid. This basis of functions is special in that the rougher the function, the more basis functions are needed to represent it. Hence this basis provides a parsimonious parameterization of sets of moderately smooth and homogeneously smooth curves (as exemplified in Sect. 4.1). There are two meanings of the term

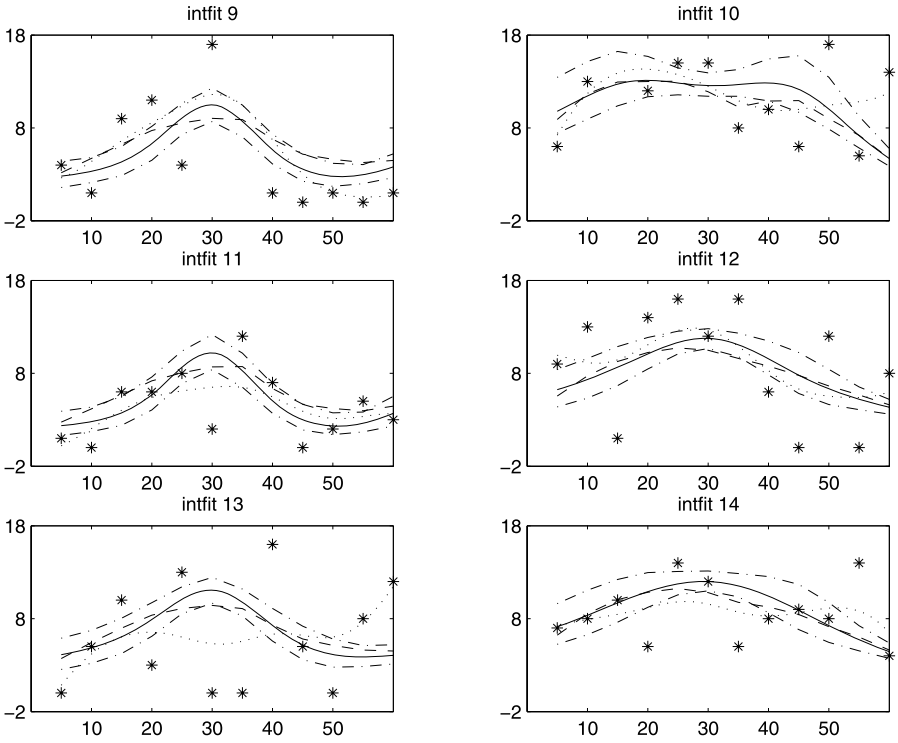

Fig. 10 The six intensity functions of War rats estimated based on individual local linear smoothing (*dotted lines*), PACE (*dashed lines*) and Bayesian FPCA (*solid lines*) together with credible sets (*dashed-dotted lines*). *Stars* mark observations ≤ 18

homogeneity of smoothness of curves: one *within* single curves, often addressed as global smoothness, and another one *across* curves. For the generative model to work well, some homogeneity of smoothness across curves is required because the generative model essentially is a multivariate multiple regression model. Non-homogeneity of smoothness within curves like spikiness requires many basis functions to represent the individual curve in the model, thus inducing a high dimensional parameterization. If furthermore roughness (like spikes) occurs at different locations, the inhomogeneity within each curve also causes inhomogeneity across curves, and the Demmler–Reinsch(-like) basis (introduced to extend the variational algorithm to *functional* PCA) does not provide a useful parameterization any more.

These aspects of modelling which were already discussed for Gaussian data have to be kept in mind when applying the model with less informative binary or count data. In particular, the required homogeneity of smoothness of curves may render the identifiability of curves more difficult and thus pose a problem of insufficient sample sizes (as illustrated in Sect. 3.1). Furthermore, the dimensionality of parameters depending on the degree of roughness needed to be represented in the model carries over to the dimensionality of the matrix of points of expansion needed to approximate the log-likelihood function. Thus the information in the data may be further diluted

and the estimation be destabilized. This is, to some extent, in contrast to the previous point because differently smooth curves are more easily identified (as in the example in Sect. 3.2).

With the model set up appropriately and the variational algorithm run with sufficiently informative data, the procedure works fast and reliably. The structures in the data (mean function and eigenfunctions) are identified even if the individual curves cannot be reconstructed accurately. Beyond point estimates (approximate) posterior distributions are obtained conditionally on the empirically chosen model parameters. For Poisson data, the proposed procedure up to now is the only (empirical) Bayesian approach to FPCA if the data are sparse so that individual curves cannot be fitted separately.

Comparisons with PACE (Hall et al. 2008) demonstrate that for sufficiently informative data sets the performance of the two methods is similar. However, particularly for less informative data sets, the Bayesian approach tends to yield smoother fits of individual curves due to the use of the Demmler–Reinsch-like basis functions. For large sample sizes ($N_m \geq 100$) the computing time of PACE exceeds considerably the computing time of the variational algorithm.

5.2 Non-functional PCA

The approximation of the log-likelihood based on working observations of course also applies to PCA in one-parameter exponential families where the columns of the matrix A in (2) directly constitute eigenvectors without any functional interpretation, that is, without modelling $A = R_d G$ (Bishop 1999a, 1999b). Examples of logistic non-functional PCA with real data are given by Schein et al. (2003). The second order Taylor expansion is particularly useful if the values of the latent factor in the matrix s are assumed to be realizations of Gaussian variables. This is appropriate in FPCA but may look different in other contexts. Discrete latent priors were suggested in extensions of independent component analysis to data from exponential families especially in the machine learning community (for example, Kabán and Girolami 2001; Sajama and Orlitsky 2004).

Appendix

In this technical appendix, some formulae are summarized in order to make the description of the algorithm self-contained. Derivations of the formulae can be found in the cited literature, mainly in Wahba (1990).

Let 1_N denote an N-dimensional vector of ones, I_N an N-dimensional identity matrix, and B^T the transpose of a matrix B.

A.1 Spline interpolation

Let two grids d and d_m be written as column vectors $d = (t_1, \ldots, t_N)^T \in \mathbb{R}^N$, $d_m = (t_{1m}, \ldots, t_{N_m m})^T \in \mathbb{R}^{N_m}$. Let further h_d denote the vector of function values of

a function h at d, $h_d = (h(t_1), \ldots, h(t_N))^T \in \mathbb{R}^N$. Then the vector $\widehat{h}_{d_m}$ of function values at d_m of the interpolation spline given h_d, $\widehat{h}_{d_m} = (h_{I(h_d)}(t_{1m}), \ldots, h_{I(h_d)}(t_{N_m m}))^T \in \mathbb{R}^{N_m}$ is given by

$$\widehat{h}_{d_m} = IP_m h_d. \tag{7}$$

$IP_m \in \mathbb{R}^{N_m \times N}$ is built from matrices $V_d = (1_N\ d)$, $V_{d_m} = (1_{N_m}\ d_m)$, and $M(d,d)$, $M(d_m,d)$ (specified in (11)) as

$$IP_m = V_{d_m} C + M(d_m, d) B \tag{8}$$

with

$$C = \left(V_d^T M(d,d)^{-1} V_d\right)^{-1} V_d^T M(d,d)^{-1}, \tag{9}$$

$$B = M(d,d)^{-1}(I_N - V_d C) \tag{10}$$

(compare Wahba 1990, Chap. 1 or Kimeldorf and Wahba 1971). The $N_m \times N$-matrix $M(d_m, d)$ is given by

$$M(d_m,d) = \frac{1}{12}\left[E(d_m,d) - W_{d_m} E^0(d) - E^0(d_m)^T W_d^T + W_{d_m} E^0 W_d^T\right], \tag{11}$$

where $W_d = (1_N - d\ d) \in \mathbb{R}^{N\times 2}$, $W_{d_m} = (1_{N_m} - d_m\ d_m) \in \mathbb{R}^{N_m \times 2}$, $E^0 = \left[\begin{smallmatrix} 0 & 1 \\ 1 & 0 \end{smallmatrix}\right] \in \mathbb{R}^{2\times 2}$,

$$E^0(d) = \begin{bmatrix} |t_1|^3 & \cdots & |t_N|^3 \\ |1-t_1|^3 & \cdots & |1-t_N|^3 \end{bmatrix} \in \mathbb{R}^{2\times N}, \tag{12}$$

$$E^0(d_m) = \begin{bmatrix} |t_{1m}|^3 & \cdots & |t_{N_m m}|^3 \\ |1-t_{1m}|^3 & \cdots & |1-t_{N_m m}|^3 \end{bmatrix} \in \mathbb{R}^{2\times N_m}. \tag{13}$$

$M(d,d)$ is obtained in the same way replacing d_m by d. (Equation (11) is the matrix version of (2.4.25) in Wahba (1990, p. 34) which, in turn, refers to her equation (2.4.10) on p. 31. In Wahba's notation, the values $d = 1$, $m = 2$, $\theta_{2,1} = 1/12$, $s_1 = 0$, $s_2 = 1$, $p_1(s) = 1 - s$, $p_2(s) = s$ are used. Note that s_1, s_2 have to be chosen such that they are not elements of one of the grids.)

The $N \times r_Q$-matrix Q_d is obtained as follows: Let Q_{r_Q} denote the $r_Q - 2$ orthonormal eigenvectors of B given in (10) corresponding to the $r_Q - 2$ smallest (!) positive eigenvalues. Let further U_d denote a $2 \times N$-matrix whose columns are given by the two orthonormal eigenvectors of $V_d V_d^T$ corresponding to the two largest eigenvalues. Then $Q_d = (U_d\ Q_{r_Q})$. Similarly, the $N \times r_R$-matrix R_d is built as $R_d = (U_d\ Q_{r_R})$. For a motivation and interpretation as PCA of interpolation splines see van der Linde (2003).

A.2 Variational algorithm for Gaussian observations

Below the variational algorithm for Gaussian observations

$$y_m | \delta, G, s_m, \sigma^2 \sim N\left(Q_m \delta + R_m G s_m, \sigma^2 I_{N_m}\right)$$

with $Q_m = IP_m Q_d$ and $R_m = IP_m R_d$ is described. In the sequel, we adopt the convention of denoting an expectation with respect to the current q by brackets $\langle\,\rangle$. Updating iteratively according to (4), the following component distributions are obtained.

– Induced by independence in the prior

$$q(S) = \prod_{m=1}^{M} q(s_m)$$

and

$$s_m \sim N(\mu_{s_m}, \Sigma_{s_m}),$$

$$\Sigma_{s_m} = \left(\left\langle \frac{1}{\sigma^2} \right\rangle \left\langle G^T R_m^T R_m G \right\rangle + I_K\right)^{-1}, \tag{14}$$

$$\mu_{s_m} = \left\langle \frac{1}{\sigma^2} \right\rangle \Sigma_{s_m} \left\langle G^T \right\rangle R_m^T \big(y_m - Q_m \langle \delta \rangle\big). \tag{15}$$

– Furthermore,

$$\delta \sim N(\mu_\delta, \Sigma_\delta),$$

$$\Sigma_\delta = \left(\beta^{-1} I_{r_Q} + \left\langle \frac{1}{\sigma^2} \right\rangle \sum_{m=1}^{M} Q_m^T Q_m\right)^{-1}, \tag{16}$$

$$\mu_\delta = \left\langle \frac{1}{\sigma^2} \right\rangle \Sigma_\delta \sum_{m=1}^{M} Q_m^T \big(y_m - R_m \langle G \rangle \langle s_m \rangle\big). \tag{17}$$

– For $G = (\gamma_{1,\ldots,}\gamma_K)$ one has

$$q(G) = \prod_{k=1}^{K} q(\gamma_k)$$

with

$$\gamma_k \sim N(\mu_{\gamma_k}, \Sigma_{\gamma_k})$$

and

$$\Sigma_{\gamma_k} = \left(\sum_{m=1}^{M} \big(s_{km}^2\big) R_m^T R_m + \left(\frac{1}{\sigma_k^2}\right) I_{r_R}\right)^{-1}, \tag{18}$$

$$\mu_{\gamma_k} = \Sigma_{\gamma_k} \sum_{m=1}^{M} R_m^T (s_{km} y_{mk}), \tag{19}$$

where $y_{mk} = y_m - Q_m \delta - R_m G_{-k} s_{-km}$ with G_{-k} denoting G without the kth column and s_{-km} denoting s_m without the kth row.

– Induced by independence in the prior

$$q(\sigma_A) = \prod_{k=1}^{K} q\left(\sigma_k^2\right),$$

and

$$\sigma_k^2 \sim IG\left(\alpha_{A0} + \frac{r_R}{2},\ \beta_{A0} + \frac{\|\gamma_k\|^2}{2}\right). \tag{20}$$

$$\sigma^2 \sim IG\left(\alpha_0 + \frac{1}{2}\sum_{m=1}^{M} N_{d_m},\ \widetilde{\beta}\right), \tag{21}$$

$$\begin{aligned}\widetilde{\beta} = \beta_0 + \frac{1}{2}\sum_{m=1}^{M}\big[&y_m^T y_m + \langle \delta^T Q_m^T Q_m \delta\rangle + \mathrm{tr}\big(\langle G^T R_m^T R_m G\rangle \langle s_m s_m^T\rangle\big) \\ &+ 2\langle\delta^T\rangle Q_m^T R_m \langle G\rangle\langle s_m\rangle - 2y_m^T R_m\langle G\rangle\langle s_m\rangle - 2y_m^T Q_m\langle\delta\rangle\big]\end{aligned} \tag{22}$$

where tr denotes the trace of a matrix.

A.3 Adaptation to non-Gaussian observations

For non-Gaussian observations, the variational algorithm builds on working observations

$$w_m\left(f_{md}^0\right)|\delta, G, s_m \sim N\left(Q_m\delta + R_m G s_m,\ D\left(f_{md}^0\right)\right).$$

Transformation with $(D(f_{md}^0))^{-1/2}$ yields

$$\widetilde{y}_m|\delta, G, s_m \sim N\left(\widetilde{Q}_m\delta + \widetilde{R}_m G s_m,\ I_{N_m}\right)$$

for $\widetilde{y}_m = (D(f_{md}^0))^{-1/2} w_m(f_{md}^0)$, $\widetilde{Q}_m = (D(f_{md}^0))^{-1/2} Q_m$, $\widetilde{R}_m = (D(f_{md}^0))^{-1/2} R_m$, and the variational algorithm for Gaussian observations can be applied with $\sigma^2 = 1$, omitting the update (15) (and (16)).

References

Achcar, J.A., Coelho-Barros, E.A., Martinez, E.Z.: Statistical analysis for longitudinal counting data in the presence of a covariate considering different "frailty" models. Braz. J. Probab. Stat. **22**, 183–205 (2008)

Behseta, S., Kass, R.E., Wallstrom, G.L.: Hierarchical models for assessing variability among functions. Biometrika **92**, 419–434 (2005)

Bishop, C.M.: Bayesian PCA Advances in Neural Information Processing Systems, vol. 11, pp. 382–388. MIT Press, Cambridge (1999a)

Bishop, C.M.: Variational principal components. In: Proceedings Ninth International Conference on Artificial Neural Networks, ICANN'99, IEE, pp. 509–514 (1999b)

Braun, M., McAuliffe, J.: Variational inference for large-scale models of discrete choice (2008). arXiv:0712.2526v3

Collins, M., DasGupta, S., Schapire, R.E.: A generalization of PCA to the exponential familes. In: Advances in Neural Information Processing Systems, vol. 15, pp. 617–624. MIT Press, Cambridge (2002)

Green, P.J., Silverman, B.W.: Nonparametric Regression and Generalized Linear Models. Chapman & Hall, London (1994)

Hall, P., Müller, H.-G., Yao, F.: Modeling sparse generalized longitudinal observations with latent Gaussian processes. J. R. Stat. Soc. B **70**, 703–723 (2008)

Jaakkola, T., Jordan, M.: A variational approach to Bayesian logistic regression models and their extensions. In: Proc. 6th Int. Workshop on Artificial Intelligence and Statistics, Ft Lauderdale, Florida (1997)

James, G.M., Hastie, T.J., Sugar, C.A.: Principal component models for sparse functional data. Biometrika **87**, 587–602 (2000)

Kabán, A., Bingham, E.: Factorisation and denoising of 0–1 data: a variational approach. Neurocomputing **71**, 2291–2308 (2008)

Kabán, A., Girolami, M.: A combined latent class and trait model for the analysis and visualization of discrete data. IEEE Trans. Patttern Anal. Mach. Intell. **23**, 859–872 (2001)

Kimeldorf, G., Wahba, G.: Some results on Tchebycheffian spline functions. J. Math. Anal. Appl. **33**, 82–95 (1971)

van der Linde, A.: PCA-based dimension reduction for splines. J. Nonparametr. Stat. **15**, 77–92 (2003)

van der Linde, A.: Variational Bayesian functional PCA. Comput. Stat. Data Anal. **53**, 517–533 (2008)

McCullagh, P., Nelder, J.A.: Generalized Linear Models. Chapman & Hall, London (1983)

Ramsay, J.O., Silverman, B.W.: Applied Functional Data Analysis: Methods and Case Studies. Springer, New York (2002)

Sajama, Orlitsky, A.: Semi-parametric exponential family PCA. In: Advances in Neural Information Processing Systems, vol. 17, NIPS 2004, Vancouver, 13–18 Dec. 2004

Schein, A.I., Saul, L.H., Ungar, A.: A generalised linear model for principal component analysis of binary data. In: Proc. 9th Int. Workshop on Artificial Intelligence and Statistics, Key West, Florida (2003)

Smith, J.A.: Statistical modelling of daily rainfall occurrences. Water Resour. Res. **23**, 885–893 (1987)

Tipping, M.E.: Probabilistic visualisation of high-dimensional binary data. In: Advances in Neural Information Processing Systems, vol. 11, pp. 592–598. MIT Press, Cambridge (1999)

Wahba, G.: Spline Models for Observational Data. SIAM, Philadelphia (1990)

Wedel, M., Kamakura, W.A.: Factor analysis with (mixed) observed and latent variables in the exponential family. Psychometrika **66**, 515–530 (2001)