A LENGTH PENALIZED PROBABILISTIC PRINCIPAL CURVE ALGORITHM WITH APPLICATIONS TO HANDWRITTEN DIGITS AND PHARMACOLOGIC COLON IMAGING

BY HUAN CHEN*, ETHEL WELD† CRAIG HENDRIX† BRIANCAFFO*

*Johns Hopkins Bloomberg School of Public Health** *and*
Johns Hopkins Medicine†

615 N WOLFE ST., BALTIMORE, MD 21205

The classical Principal Curve algorithm was developed as a nonlinear version of principal component analysis to model curves. However, existing principal curve algorithms with classical penalties, such as smoothness or ridge penalties, lack the ability to deal with complex curve shapes. In this manuscript, we introduce a robust and stable length penalty which solves issues of unnecessary curve complexity, such as the self-looping, that arise widely in principal curve algorithms. A novel probabilistic mixture regression model is formulated. A modified penalized EM(Expectation Maximization) Algorithm was applied to the model to obtain the penalized MLE. Two applications of the algorithm were performed. In the first, the algorithm was applied to the MNIST dataset of handwritten digits to find the centerline, not unlike defining a TrueType font. We demonstrate that the centerline can be recovered with this algorithm. In the second application, the algorithm was applied to construct a three dimensional centerline through single photon emission computed tomography images of the colon arising from the study of pre-exposure prophylaxis for HIV. The centerline in this application is crucial for understanding the distribution of the antiviral agents in the colon for HIV prevention. The new algorithms improves on previous applications of principal curves to this data.

1. Introduction. A commonly encountered problem in biomedical image processing is trying to find the centerlines of anatomical structures, such as blood vessels, neurons or colons. Researchers in the computer vision and medical image processing literature have developed various methods to estimate centerlines, including: virtual colonoscopy and techniques for the localization of polyps [1, 2, 3, 4, 5]. A related, well researched topic includes the use of Bezier curves ad B-splines [6]. Techniques considering the image as a connected graph represent another direction [7, 8, 4, 9, 10].

Curve fitting is less well represented in the statistical literature, especially when compared to the vast literature in non-parametric function estimation. Notably, [11] introduced principal curves, which were defined via a self-consistency property, where for fitting they applied an alternating minimization procedure. In [12], the author proposed an alternative principal curve definition via local splines based on mixture models and applied the EM algorithm for estimation. In [13], a clustering algorithm based on principal curves was proposed. In [14], a modified version of Hastie's original principal curve algorithm was suggested that slowly increased curve complexity and allowed the user to take pixel intensities into account and to constrain the starting, ending and interior points. This greatly improved the practical performance of the traditional principal curve algorithm for a colon imaging application that we continue to investigate herein.

Principal curve algorithms have also been applied in other areas, including physics [11, 15], natural language processing [16, 17], geology [18, 13, 19, 20], natural sciences [21, 20] and bio-medical studies [22, 14]. They are often useful over connected graphs, for example, in settings where the structure is not contiguous, there is noise and when sampling assumptions are needed. Our colon imaging example is one such setting.

In this article, we focus on principal curves for this and more general settings. We specifically address the shortcomings of [14] and propose a novel principal curve algorithm, which we call the **probabilistic length penalized principal curve**. This algorithm builds on, yet differs from existing ones by: (1) having a length penalty, which solves a self-looping problem often encountered in principal curve algorithms; (2) formulating a length penalized probabilistic model for the principal curve with constrained conditions; (3) utilizing an efficient penalized EM algorithm accommodating constraint conditions to estimate the parameters in the model.

The structure of the paper is organized as below. In section 2, we illustrate and give examples about the drawbacks of the current penalty used in principal curve and introduce a new length penalty. In section 3, The probabilistic principal curve is introduced and a modified EM algorithm to estimate the parameters for the curve is derived. In section 4, we apply the probabilistic length penalized principal curve model to the MNIST dataset, to illustrate how the length penalty works, and how the algorithm performs in a highly contrived setting. In section 5, the algorithm is applied to the data from a pharmacologic colon imaging study for the study of the kinetics of microbicide candidates. In section 6, a discussion of this new algorithm is given along with and future directions.

2. Primary application. Our primary motivation for developing this algorithm was to study the distribution of anti-microbial treatments in the human colon as a pre-exposure prophylaxis against the spread of the the human immunodeficiency virus (HIV) for receptive anal intercourse with one infected partner. Such treatments are potentially useful for preventing HIV transmission by being more behaviorally congruent [23, 24, 25] than oral dosing, which is highly effective, yet also prone to issues of adherence.

To understand the potential efficacy of the treatment, its distribution and the kinetics within the lumen of the colon after anal intercourse needs to be studied. However, the complexity of the physical forces prevents theoretical or computer simulation analysis of the microbial agent, and thus it is studied in vivo using imaging. Our data comes from one such experiment. The primary aim of this investigation is to consider the problem of estimating the distributional properties of the treatment after anal intercourse. These are typically derived as summaries of an fitted centerline to the image of the antiviral product vehicle [24, 26].

For imaging, a single photon emission computed tomography (SPECT) scanner was used to visualize the distribution of a microbicide candidate mixed with a radiotracer. The imaging system resconstructs an image of the tracer from its emissions using tomogrpahic reconstruction. Image intensities represent the distribution of the tracer at that location. The tracer image then surves as a surrogate for the microbicide. The treatment was inserted and distributed via physical forces consistent with those of anal intercourse.

The SPECT imaging data is represented as $128 \times 128 \times 128$ array. Each voxel (three dimensional pixel) represents a $3.45mm^3$ physical area. Accompanying each pixel is an intensity value representing the concentration of the tracer at that location. The absolute value of the concentration varies by several factors, such as attenuation, scatter and blur and other patient characteristics and aspects of the tomographic reconstruction. Therefore, intra-subject relative values are of more interest than absolute ones.

The study was approved by the Johns Hopkins Institutional Review Board and informed written consent was given.

3. A length penalty principal curve algorithm.

3.1. *Principal curve algorithms.* Let f, our target of estimation, be a curve in 2D or 3D space. That is, a function from $\mathscr{R}$ to $\mathscr{R}^2$ or $\mathscr{R}^3$. We observe, x_i, a collection of points in $\mathscr{R}^d$, where d matches the range dimension of f. The x_i are modeled as a noisy realization of f; we do not observe, λ_i, where, say, $x_i = f(\lambda_i) + \varepsilon_i$. If the λ_i were observed, for example if f was a trajectory of a particle and λ_i and x_i were the time and location respectively, then the problem could be solved by ordinary scatterplot smoothing methods.

The classical principal curve algorithm [11] used smoothing splines to minimize the term:

$$D^2(\boldsymbol{f},\lambda) = \sum_{i=1}^{n} ||\boldsymbol{x}_i - \boldsymbol{f}(\lambda_i)||^2 + \mu \int ||\boldsymbol{f}''(\lambda)||^2 d\lambda,$$

among functions, f, in the estimation space of splines restricted to f' being absolutely continuous and $f'' \in L_2$. An block relaxation algorithm was derived to minimize this loss function. The classical principal curve algorithm is as follows.

Step 1: Given $\boldsymbol{f}$, minimize $D^2(\boldsymbol{f},\lambda)$ over λ_i; this step projects points onto the curve.
Step 2: Given λ_i, apply the cubic spline smoother to components of f separately, with penalty μ; this step fits a cubic spline estimate to f.

Of course, any smoothing approach could be used in Step 2, such as a p-spline basis approach [27]. For example, suppose that the basis function for the spline can be represented as $[B_1(t), B_2(t), \ldots, B_s(t)]^T$, then the penalty term is $\mu \int ||f''(\lambda)||^2 d\lambda = \mu \boldsymbol{\beta}^T \boldsymbol{\Omega} \boldsymbol{\beta}$, where $\Omega_{jk} = \int B_j''(t) B_k''(t) dt$. A standard ridge regression penalty is specifically when $\Omega_{ij} = I(i = j)$, where I is an indicator function. Modern generalizations on this approach utilize random effect mixed model BLUP estimation [28, 29, 30]. In this method, no grid search or cross validation for the penalty is needed, as the smooth penalty can be written as the ratio between the variance of random effect and the error variances. To estimate this quantity, ML or REML can be used, with best linear unbiased predictors (BLUP) then used to estimate the smoother.

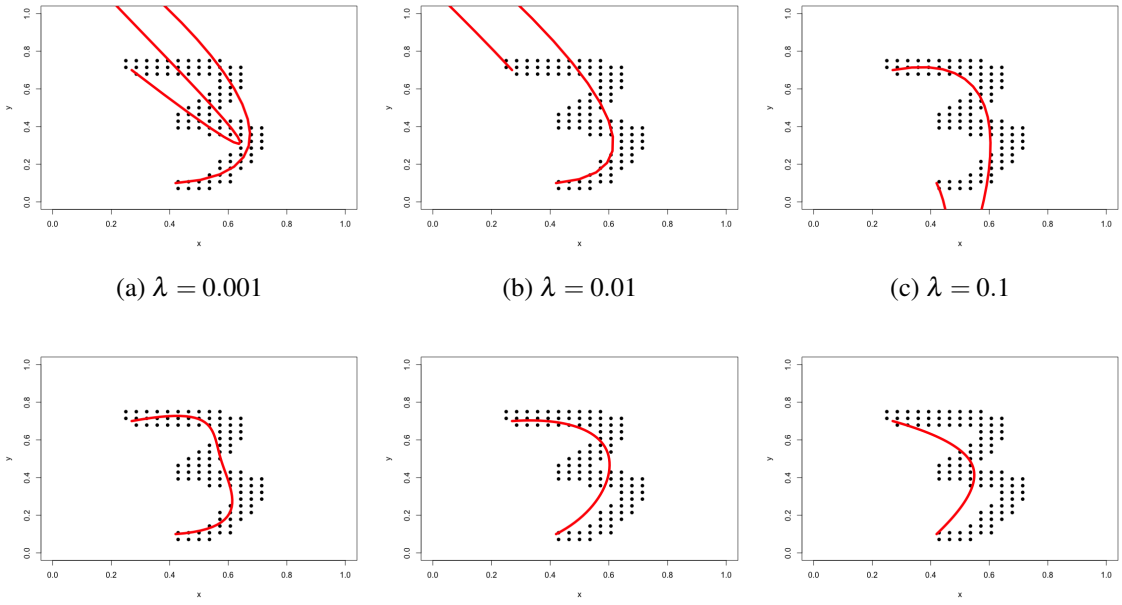
When using traditional or modern spline or penalized function estimation in Step 2 of the principal curve algorithm, the penalty only controls the smoothness of the curve, not the length. However, unlike a traditional scatterplot smoothing problem, the principal curve algorithm also estimates λ in Step 1. Complexity in the curve to fit complex real features can be accompanied by curves that loop in on themselves and make unnatural bends to reduce error.

The experiments in Figure 1 demonstrate an example on a handwritten digit 3 from the MNIST database [31]. The subpanels show various values of the penalty for a cubic spline smoother. Note that very smooth functions are clearly biased and not a useful representation of the underlying structure. The more complex functions [Panels (a), (b), (c)] show unnatural bends in the curves to minimize projection distances. This occurs, as principal curves are non-unique, curves that violate our intuitive notion of a solution are, in fact, often good minimizers of the loss function from Step 2 of the algorithm. Because the algorithm chooses the location of the λ parameters in addition to the curve fit, the fitted curve may be paying little to no penalty for sharp turns or curve components completely outside of the data. As we have seen, modifying penalties or degrees of freedom in the smoother alone is not a viable solution. Nor is changing the kind of smoother or penalty, as we show below.

Figure 2 repeats the study using a ridge penalty. An example of the function looping in on itself unnecessarily is highlighted in Panel (c). The change in smoothing approach slightly improves performance, but continues to differ from an intuitive fit to the data.

We show the results of the length penalty (before introducing the algorithm) in Figure 3. In Panels (b), (c) and (d), it recovers an underlying representation that mirrors our intuition. Of course, it can overfit [Panel (a)] or underfit [Panels (e) and (f)], so that choice of the penalty remains does remain an important component of the algorithm. Nonetheless, the penalty prevents the creation of unnatural bends and self loops elsewhere in the curve to fit local increases in curvature and creates a form of robustness to the degree of smoothing in the second step of the algorithm.

3.2. *Length Penalty.* A length penalty is a potential solution to allow for additional flexibility of the curve beyond those provided by traditional smoothing parameters. Abstractly, a length penalty can be thought of as a form of prior information attempting to force the principal curve to better adhere to our intuitive notion of what constitutes an acceptable curve fit.



(a) $\lambda = 0.001$ (b) $\lambda = 0.01$ (c) $\lambda = 0.1$

(d) $\lambda = 1$ (e) $\lambda = 10$ (f) $\lambda = 100$

Fig 1: The performance of smoothness penalties on a example image. Panels (a) and (b) both use relatively small penalties where the fitted curve tends to be long and does not capture the ideal shape of the written digit. Panels (c), (d), (e), (f) have a relatively large penalty, but the fitted curve does not capture the curvature necessary for an intuitive fit. The figure highlights that tuning the smoothness penalty is not the fundamental problem in principal curves.

Consider a parameterized curve function $\boldsymbol{f}(\lambda) = \{f^x(\lambda), f^y(\lambda), f^z(\lambda)\}, \lambda \in [0, 2\pi]$. The three coordinates with respect to the x, y, z axis are

$$\begin{aligned} x(\lambda_k) &= f^x(\lambda_k) = \boldsymbol{B}(\lambda_k)^T \boldsymbol{\beta}^x, \\ y(\lambda_k) &= f^y(\lambda_k) = \boldsymbol{B}(\lambda_k)^T \boldsymbol{\beta}^y, \\ z(\lambda_k) &= f^z(\lambda_k) = \boldsymbol{B}(\lambda_k)^T \boldsymbol{\beta}^z, \end{aligned}$$

where B forms a set of basis for the splines. Any typical spline basis sets can be used; we use B-Splines. Let $\boldsymbol{\lambda}$ be a sequence of non-decreasing real numbers $(\lambda_i \leq \lambda_{i+1})$ such that

$$\lambda_0 \leq \lambda_1 \leq \cdots \leq \lambda_{N+1}.$$

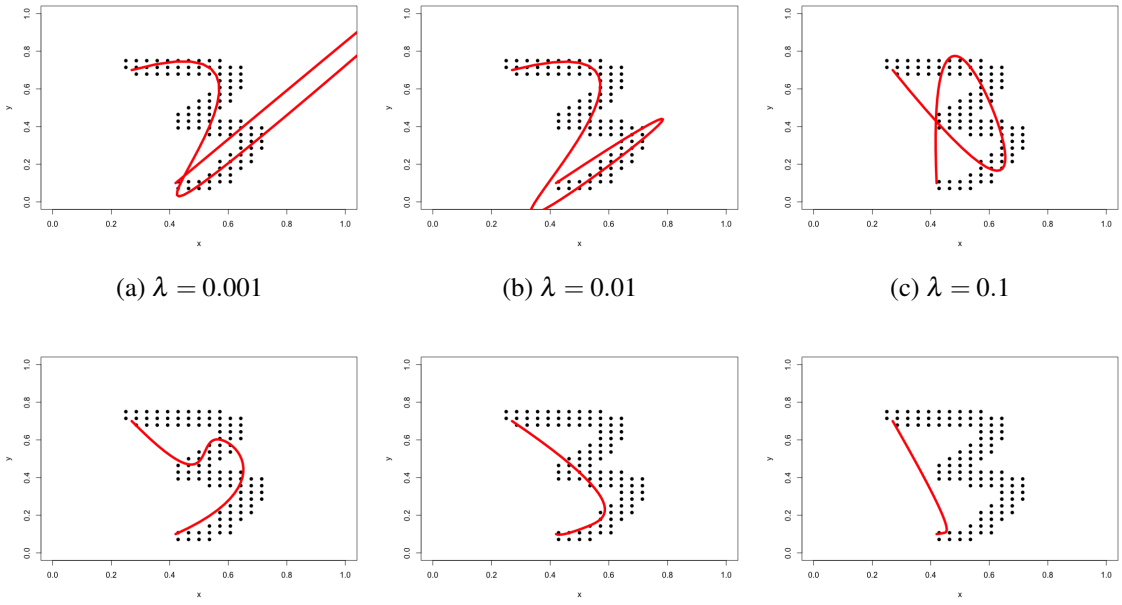
Defined the augmented knot set as:

$$\lambda_{-(m-1)} = \ldots = \lambda_0 \leq \lambda_1 \leq \ldots \leq \lambda_N \leq \lambda_{N+1} = \ldots \lambda_{N+m},$$

where we have appended lower and upper boundary knots, λ_0 and λ_N, $m-1$ times. Here, slightly abusing notation, we reset the index so that the $N+2m$ augmented knots t_i are now indexed by $i = 0, \ldots, N+2m-1$.

To define a standard B-spline basis, for each of the augmented knots, $\lambda_i, i = 0, \ldots, N+2m-1$, recursively define a set of real-valued functions, $B_{i,j}(x)$, so that:

$$B_{i,0}(x) = \begin{cases} 1 & \text{if } \lambda_i \leq x < \lambda_{i+1} \\ 0 & \text{otherwise} \end{cases},$$



(a) $\lambda = 0.001$ (b) $\lambda = 0.01$ (c) $\lambda = 0.1$

(d) $\lambda = 1$ (e) $\lambda = 10$ (f) $\lambda = 100$

Fig 2: The performance of ridge penalties. In Panel (a), (b), when the penalty is small, the shape of number 3 is not captured well, the coefficient for the spline tends to take large values in order to maximize the likelihood. In Panel (c), the penalty parameter does not capture the shape of the 3. In Panels (d), (e), (f), with large penalties, the shape is completely missed.

$$B_{i,j+1(x)} = \alpha_{i,j+1}(x)B_{i,j}(x) + [1 - \alpha_{i+1,j+1}(x)]B_{i+1,j}(x),$$

where

$$\alpha_{i,j}(x) = \begin{cases} \frac{x-\lambda_i}{\lambda_{i+j}-\lambda_i} & \text{if } \lambda_{i+j} \neq \lambda_i \\ 0 & \text{otherwise} \end{cases}.$$

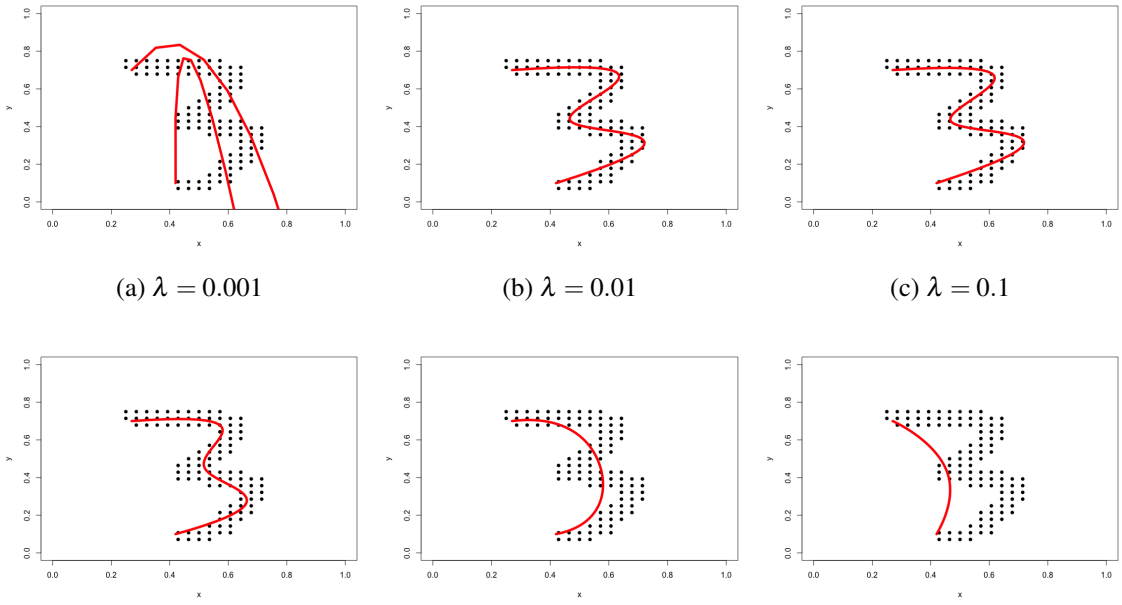
(For the above computation, 0/0 is defined as 0.) Suppose that the degree of our proposed curve is p, then abbreviate $B_{i,p}(x)$ as $B_i(x)$, so that the spline basis can be written as:

$$\boldsymbol{B}(x) = [B_0(x), B_1(x), \ldots, B_{N+2m-1}(x)]^T.$$

Under this definition of the B-spline basis, the arc length of the 3D curve can be computed as:

$$\begin{aligned} &\int_0^{2\pi} \sqrt{\left(\frac{df^x(\lambda)}{d\lambda}\right)^2 + \left(\frac{df^y(\lambda)}{d\lambda}\right)^2 + \left(\frac{df^z(\lambda)}{d\lambda}\right)^2} d\lambda \\ =&\int_0^{2\pi} \sqrt{\boldsymbol{\beta}_x^\top \tilde{\boldsymbol{B}}(\lambda)\tilde{\boldsymbol{B}}^\top(\lambda)\boldsymbol{\beta}_x + \boldsymbol{\beta}_y^\top \tilde{\boldsymbol{B}}(\lambda)\tilde{\boldsymbol{B}}^\top(\lambda)\boldsymbol{\beta}_y + \boldsymbol{\beta}_z^\top \tilde{\boldsymbol{B}}(\lambda)\tilde{\boldsymbol{B}}^\top(\lambda)\boldsymbol{\beta}_z} d\lambda \\ \approx&\frac{2\pi}{K}\sum_{k=1}^{K} \sqrt{\boldsymbol{\beta}_x^\top \tilde{\boldsymbol{B}}(\lambda_k)\tilde{\boldsymbol{B}}^\top(\lambda_k)\boldsymbol{\beta}_x + \boldsymbol{\beta}_y^\top \tilde{\boldsymbol{B}}(\lambda_k)\tilde{\boldsymbol{B}}^\top(\lambda_k)\boldsymbol{\beta}_y + \boldsymbol{\beta}_z^\top \tilde{\boldsymbol{B}}(\lambda_k)\tilde{\boldsymbol{B}}^\top(\lambda_k)\boldsymbol{\beta}_z}, \end{aligned}$$

where $\tilde{\boldsymbol{B}}(\lambda)$ is the gradient of $\boldsymbol{B}$ with respect to λ, $\tilde{\boldsymbol{B}}(\lambda) = [\frac{d}{d\lambda}B_0(\lambda), \frac{d}{d\lambda}B_1(\lambda), \ldots, \frac{d}{d\lambda}B_{N+2m-1}(\lambda)]^T]$, and β_w (for $w = x, y, z$) are the associated coefficients.



(a) $\lambda = 0.001$ (b) $\lambda = 0.01$ (c) $\lambda = 0.1$

(d) $\lambda = 1$ (e) $\lambda = 10$ (f) $\lambda = 100$

Fig 3: The performance of length penalties. In Panel (a), the penalty is too small, making the fitted curve to complex. In Panels (b), (c), (d), the shape of 3 is captured well even though the penalty varies greatly: highlighting the robustness of the algorithm to penalty choice. In Panels (e), (f), with a large length penalty, the algorithm underfits the curve.

We propose adding the length approximation as a penalty. However, because of the non-convexity of the length with respect to $\beta_x, \beta_y, \beta_z$, this penalty is numerically inconvenient for maximization. Thus, we relax the penalty by dropping the square root, which makes the penalty quadratic in β_x, β_y and β_z as follows:

$$\begin{aligned}&\frac{2\pi}{K}\sum_{k=1}^{K}[\boldsymbol{\beta}_x^\top \tilde{\boldsymbol{B}}(t_k)\tilde{\boldsymbol{B}}^\top(t_k)\boldsymbol{\beta}_x + \boldsymbol{\beta}_y^\top \tilde{\boldsymbol{B}}(t_k)\tilde{\boldsymbol{B}}^\top(t_k)\boldsymbol{\beta}_y + \boldsymbol{\beta}_z^\top \tilde{\boldsymbol{B}}(t_k)\tilde{\boldsymbol{B}}^\top(t_k)\boldsymbol{\beta}_z]\\ =&\boldsymbol{\beta}_x^\top \tilde{\Omega}_B \boldsymbol{\beta}_x + \boldsymbol{\beta}_y^\top \tilde{\Omega}_B \boldsymbol{\beta}_z + \boldsymbol{\beta}_x^\top \tilde{\Omega}_B \boldsymbol{\beta}_z.\end{aligned}$$

Our proposed length penalty is then:

$$P_{length}(\lambda) = \lambda\{\boldsymbol{\beta}_x^\top \tilde{\Omega}_B \boldsymbol{\beta}_x + \boldsymbol{\beta}_y^\top \tilde{\Omega}_B \boldsymbol{\beta}_z + \boldsymbol{\beta}_x^\top \tilde{\Omega}_B \boldsymbol{\beta}_z\},$$

where $\tilde{\Omega}_B = [\frac{2\pi}{K}\sum_{k=1}^{K}\tilde{\boldsymbol{B}}(\lambda_k)\tilde{\boldsymbol{B}}^\top(t_k)]$.

It is of interest that the form of the penalty is equivalent to minus twice the log of joint density where $\boldsymbol{\beta}_x$, $\boldsymbol{\beta}_y$ and $\boldsymbol{\beta}_z$ are independent $N(\mathbf{0}, \tilde{\Omega}_B)$. Thus, the length penalty is exactly a form of prior on the curve through the smoother coefficients.

4. Probabilistic Principal Curves. The primary difference between a probabilistic formulation of principal curve and the traditional principal curve formulation is similar to that between K-means and Gaussian Mixture Models (GMM) for clustering. In K-means, one applies a hard assignment of each point to a cluster. In contrast, in Gaussian Mixture Models, one applies a soft assignment to each data point, computing the probability of the cluster the point should belong to. In the same way, for the principal curve algorithm,

when applying the traditional principal curve algorithm, one makes a hard assignment of λ_i for each data point. For a latent variable modelling approach, one computes the probability that every 3D point belongs to each cluster. In this formulation, more local information is used compared to hard assignment.

4.1. *Probabilistic Formulation.* A mixture model for principal curves was introduced in [12]. This mode shares a similar structure to our model. However, instead of directly incorporating a nonparametric spline regression into the mixture model, the method applies a two stage method, where the mean of each mixture component was first estimated, and then a spline smooth was applied to the fitted mean. The form is inefficient in our setting, where the number of parameters is the total number of knots to be fitted in the spline for each axis. Instead, in our proposed method, the number of parameters is the degrees of freedom (which is typically smaller than the number of knots).

Let $t_1 \dots t_K$ be pre-specified cluster locations for the latent variable, which we also denote by λ_i. The typical choice for the t_k is an equally spaced grid on $[0, 2\pi]$. We further assume that the λ_i are iid and let $\pi_k = p(\lambda_i = t_k)$ be the class probabilities. Then, we assume that the observed data points in the image are independent and follow a normal distribution. Specifically, $\boldsymbol{x}_i \sim N\{\boldsymbol{f}(\lambda_i), \Sigma_{\lambda_i}\}$.

Then, the complete data likelihood for $(\boldsymbol{x}_i, \boldsymbol{\lambda}_i)$ is then given by:

$$\Pi_{i=1}^{N} \Pi_{k=1}^{K} \left[\frac{\pi_k}{(2\pi)^{3/2} |\Sigma_{\lambda_i}|^{-1/2}} e^{-\frac{1}{2}\{\boldsymbol{x}_i - \boldsymbol{f}(\lambda_i)\}^\top \Sigma_{\lambda_i}^{-1} \{\boldsymbol{x}_i - \boldsymbol{f}(\lambda_i)\}} \right]^{1_{\lambda_i = t_k}}.$$

The marginal likelihood can be calculate as the product of the marginalized individual data points,

$$p(\boldsymbol{x}) = \Pi_{i=1}^{N} \sum_{k=1}^{K} \frac{\pi_k}{(2\pi)^{3/2} |\Sigma_{\lambda_i}|^{-1/2}} e^{-\frac{1}{2}\{\boldsymbol{x}_i - \boldsymbol{f}(t_k)\}^\top \Sigma_{t_k}^{-1} \{\boldsymbol{x}_i - \boldsymbol{f}(t_k)\}}.$$

The marginal likelihood can be cumbersome to work with for a variety of reasons. For example, the inner sum does not distribute when taking logs and the simplex constraints on the π_k make direct maximization of the marginal likelihood unstable. Instead, we utilized an EM algorithm [32], which is a common solution for latent mixture model problems.

Further, for both a reduction in computational burden and increasing the stability of the estimates, we assume a simplified structure for the correlation across coordinates in that $\boldsymbol{\Sigma}_{\lambda_i} = \text{Diag}\{(\sigma_x^2 \ \sigma_y^2 \ \sigma_z^2)^\top\}$. Of course, this structure could be relaxed to capitalize on shared coordinate information, representing a possible avenue for further research.

Based on this variance-covariance structure, we have the log likelihood for $\boldsymbol{x}_i$ after conditional on λ_i is :

$$\begin{aligned} \log\{p(\boldsymbol{x}_i|\lambda_i)\} &= -\frac{3}{2}\log 2\pi - \frac{1}{2}\log\sigma_x^2 - \frac{1}{2}\log\sigma_y^2 - \frac{1}{2}\log\sigma_z^2 \\ &- \frac{1}{2\sigma_x^2}\{x_i - f^x(\lambda_i)\}^2 - \frac{1}{2\sigma_y^2}\{y_i - f^y(\lambda_i)\}^2 - \frac{1}{2\sigma_z^2}\{z_i - f^z(\lambda_i)\}^2. \end{aligned}$$

Given these building blocks, it is relatively straightforward to develop an EM algorithm for fitting. However, typical applications of the method often required constrained starting and ending points for the curve. In addition, biologically motivated constrained interior points are also often desired.

Thus, a possible improvement of the above algorithm is to manually set the starting, ending and posibly interior points of the curve. Here, we use the start and end point specification procedure developed in [14]. Suppose that $\boldsymbol{x}_0 = (x_0, y_0, z_0)$ and $\lambda_0 = 0$ and $\boldsymbol{x}_{n+1} = (x_{n+1}, y_{n+1}, z_{n+1})$ and $\lambda_{n+1} = 1$ are given. Forcing the curve through these points requires constrained least squares regression. Let $\tilde{W}$ be the basis evaluated at the constrained values of λ and let $\tilde{x}, \tilde{y}, \tilde{z}$ be vectors of the constrained values. Then the constraints can be expressed as:

$$\tilde{\boldsymbol{W}}\boldsymbol{\beta}^x = \tilde{\boldsymbol{x}}, \tilde{\boldsymbol{W}}\boldsymbol{\beta}^y = \tilde{\boldsymbol{y}}, \tilde{\boldsymbol{W}}\boldsymbol{\beta}^z = \tilde{\boldsymbol{z}},$$

with the object as

$$\begin{aligned} E[\boldsymbol{x}] &= \boldsymbol{B}\beta^{\boldsymbol{x}}, \\ E[\boldsymbol{y}] &= \boldsymbol{B}\beta^{\boldsymbol{y}}, \\ E[\boldsymbol{z}] &= \boldsymbol{B}\beta^{\boldsymbol{z}}. \end{aligned}$$

The fitted value of the constrained least squares can be expressed as

$$\begin{aligned} \hat{\boldsymbol{\beta}}_c^x &= \hat{\boldsymbol{\beta}}^x - (\boldsymbol{B}'\boldsymbol{B})^{-1}\tilde{\boldsymbol{W}}\{\tilde{\boldsymbol{W}}'(\boldsymbol{W}'\boldsymbol{W})^{-1}\}^{-1}(\tilde{\boldsymbol{W}}\hat{\boldsymbol{\beta}}^x - \tilde{\boldsymbol{x}}), \\ \hat{\boldsymbol{\beta}}_c^y &= \hat{\boldsymbol{\beta}}^y - (\boldsymbol{B}'\boldsymbol{B})^{-1}\tilde{\boldsymbol{W}}\{\tilde{\boldsymbol{W}}'(\boldsymbol{W}'\boldsymbol{W})^{-1}\}^{-1}(\tilde{\boldsymbol{W}}\hat{\boldsymbol{\beta}}^y - \tilde{\boldsymbol{y}}), \\ \hat{\boldsymbol{\beta}}_c^z &= \hat{\boldsymbol{\beta}}^z - (\boldsymbol{B}'\boldsymbol{B})^{-1}\tilde{\boldsymbol{W}}\{\tilde{\boldsymbol{W}}'(\boldsymbol{W}'\boldsymbol{W})^{-1}\}^{-1}(\tilde{\boldsymbol{W}}\hat{\boldsymbol{\beta}}^z - \tilde{\boldsymbol{z}}). \end{aligned}$$

Here, $\hat{\beta}^x, \hat{\beta}^y, \hat{\beta}^z$ are the fitted coefficients without constrains. We combine the pieces outlined up to this point in the following EM algorithm.

Step 1: Select the start point $\boldsymbol{x}_0 = (x_0, y_0, z_0)$ with $t_0 = 0$ and end point $\boldsymbol{x}_{N+1} = (x_{N+1}, y_{N+1}, z_{N+1})$ with $t_{N+1} = 2\pi$;
Step 2: Initialize the parameters, including $\pi, \sigma_x^2, \sigma_y^2, \sigma_z^2, \beta^x, \beta^y, \beta^z$
Step 3: Update the parameters iteratively:

$$\begin{aligned} \hat{\pi}^t &= \frac{\sum_{i=1}^N \theta_{ik}^{(t-1)}}{N}, \\ \hat{\sigma}_x^{2(t)} &= \frac{1}{N}\sum_{i=1}^N \sum_{k=1}^K \theta_{ik}^{(t-1)}(x_i - \boldsymbol{B}^\top(t_k)\hat{\boldsymbol{\beta}}_c^{x(t)})^2, \\ \hat{\sigma}_y^{2(t)} &= \frac{1}{N}\sum_{i=1}^N \sum_{k=1}^K \theta_{ik}^{(t-1)}(y_i - \boldsymbol{B}^\top(t_k)\hat{\boldsymbol{\beta}}_c^{y(t)})^2, \\ \hat{\sigma}_z^{2(t)} &= \frac{1}{N}\sum_{i=1}^N \sum_{k=1}^K \theta_{ik}^{(t-1)}(z_i - \boldsymbol{B}^\top(t_k)\hat{\boldsymbol{\beta}}_c^{z(t)})^2, \end{aligned}$$

where:

$$\begin{aligned} \theta_{ik}^{(t-1)} &= P(\lambda_i = t_k|\boldsymbol{x}_i) = \frac{p^{(t-1)}(\boldsymbol{x}_i|t_k)\pi_k^{(t-1)}}{\sum_k p^{(t-1)}(\boldsymbol{x}_i|t_k)\pi_k^{(t-1)}}, \\ \hat{\boldsymbol{\beta}}_c^{x(t)} &= \hat{\boldsymbol{\beta}}^{x(t)} - (\boldsymbol{B}'\boldsymbol{B})^{-1}\tilde{\boldsymbol{W}}\{\tilde{\boldsymbol{W}}'(\boldsymbol{W}_2^{(t-1)'}\boldsymbol{W}_2^{(t-1)})^{-1}\}^{-1}(\tilde{\boldsymbol{W}}\hat{\boldsymbol{\beta}}^{x(t)} - \tilde{\boldsymbol{x}}), \\ \hat{\boldsymbol{\beta}}_c^{y(t)} &= \hat{\boldsymbol{\beta}}^{y(t)} - (\boldsymbol{B}'\boldsymbol{B})^{-1}\tilde{\boldsymbol{W}}\{\tilde{\boldsymbol{W}}'(\boldsymbol{W}_2^{(t-1)'}\boldsymbol{W}_2^{(t-1)})^{-1}\}^{-1}(\tilde{\boldsymbol{W}}\hat{\boldsymbol{\beta}}^{y(t)} - \tilde{\boldsymbol{y}}), \\ \hat{\boldsymbol{\beta}}_c^{z(t)} &= \hat{\boldsymbol{\beta}}^{z(t)} - (\boldsymbol{B}'\boldsymbol{B})^{-1}\tilde{\boldsymbol{W}}\{\tilde{\boldsymbol{W}}'(\boldsymbol{W}_2^{(t-1)'}\boldsymbol{W}_2^{(t-1)})^{-1}\}^{-1}(\tilde{\boldsymbol{W}}\hat{\boldsymbol{\beta}}^{z(t)} - \tilde{\boldsymbol{z}}), \\ \hat{\boldsymbol{\beta}}^{x(t)} &= (\boldsymbol{B}^\top\boldsymbol{W}_1^{(t-1)}\boldsymbol{B} + \lambda\tilde{\boldsymbol{\Omega}}_B)^{-1}\boldsymbol{B}^\top\boldsymbol{W}_2^{(t-1)}x, \\ \hat{\boldsymbol{\beta}}^{y(t)} &= (\boldsymbol{B}^\top\boldsymbol{W}_1^{(t-1)}\boldsymbol{B} + \lambda\tilde{\boldsymbol{\Omega}}_B)^{-1}\boldsymbol{B}^\top\boldsymbol{W}_2^{(t-1)}y, \\ \hat{\boldsymbol{\beta}}^{z(t)} &= (\boldsymbol{B}^\top\boldsymbol{W}_1^{(t-1)}\boldsymbol{B} + \lambda\tilde{\boldsymbol{\Omega}}_B)^{-1}\boldsymbol{B}^\top\boldsymbol{W}_2^{(t-1)}z. \end{aligned}$$

$\tilde{\boldsymbol{\Omega}}_B = [\frac{2\pi}{K}\sum_{k=1}^K \tilde{\boldsymbol{B}}(t_k)\tilde{\boldsymbol{B}}^\top(t_k)]$, B is the basis matrix of the spline. W_1, W_2 are both weighted probability matrix used in the EM algorithm, where:

$$(\boldsymbol{W}_1^{(t-1)})_{ks} = \begin{cases} \sum_{i=1}^N \theta_{ik}^{(t-1)} & k = s \\ 0 & \text{else}, \end{cases}$$

$$(\boldsymbol{W}_2^{(t-1)})_{ij} = \theta_{ji}^{(t-1)}.$$

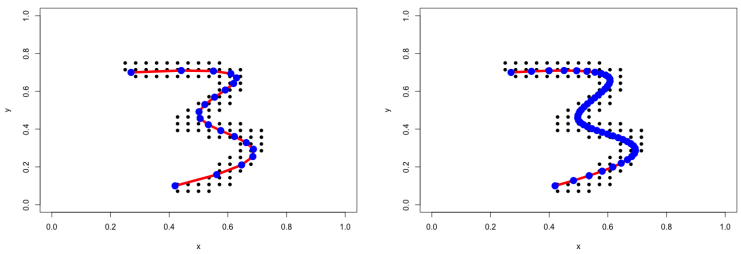
Step 4: Iterate **Step 3**, until a convergence condition is satisfied or the maximum iteration limit prespecified is reached.

Because of the ascent property of the EM algorithm, the penalized likelihood for the principal curve, $L_\lambda(f)$, will increase with each iteration. We stop the EM algorithm when the proportional increase in the likelihood reaches a pre-specified maximum number of iterations or tolerance, ε:

$$\frac{L_\lambda(f^{(t-1)}) - L_\lambda(f^{(t)})}{L_\lambda(f^{(t-1)})} < \varepsilon.$$

5. Algorithm applied to the MNIST dataset. The MNIST data forms an excellent benchmark to test the performance of the proposed algorithm. The algorithm was applied to this dataset with the aim of studying whether the intrisic shape hand written digits can be captured. Note this is dramatically different than the typical study of the MNIST dataset, where digit classification is being considered. Instead, one could think of our algorithm as attempting to create a data derived vector font from raster images of hand written digits.

We first demonstrate that the length penalty is, in fact, an adaptive algorithm that can choose the distance between knots based on the complexity of the curve itself Figure 4. For the number 3, both the upper part and lower part of it tend to be fit well, while the centerpoint is more complex. As such, our length penalized algorithm uses more knots in the sharp center point instead of the smooth upper or lower portions.

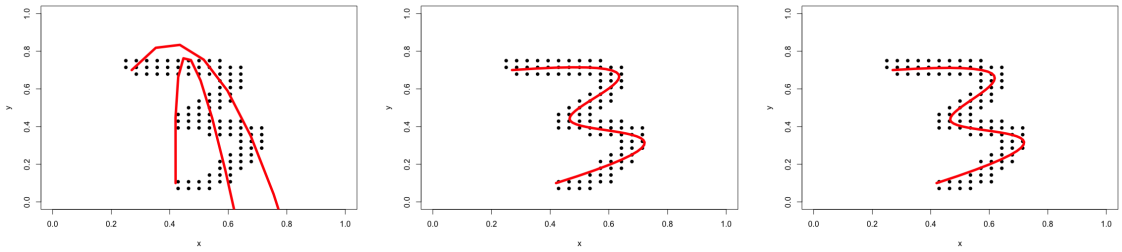


(a) 20 Knots (b) 50 Knots

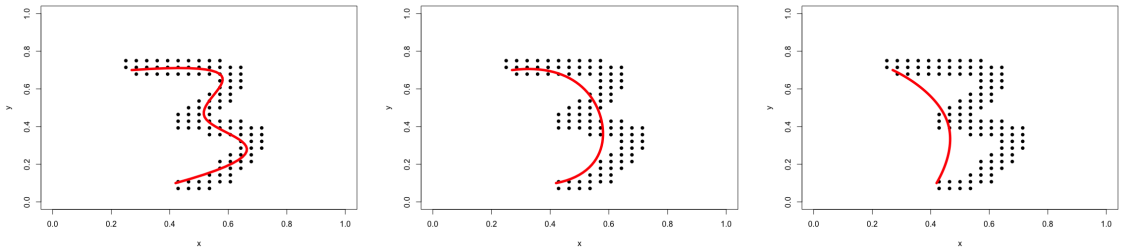
Fig 4: Self-adaptiveness of the length penalized principal curve.

The principal curve fit with different length penalties is shown in Figure 5. When the penalty was small, the fit of the principal curve appears poor, in that the curve does not follow the central path along the number. Instead, it goes through the digit arbitrarily, trying to maximize the likelihood of the mixture. However, the length penalty appears to solve this issue. When $\lambda = 0.1$, the principal curve fits the shape perfectly. In addition, the algorithm appears to be robust to the penalty value, where any penalty between 0.01 and 1 yields good results. Of course, when the penalty is dramatically increased, $\lambda = 100$, the principal curve shrinks to a line. There, the length of the curve is the shortest possible, but it ignores the shape of the data altogether.

The fitting procedure of the length penalized principal curve was also investigated, with the penalty set at $\lambda = 0.1$. With randomly initialized regression coefficients and variance components, the algorithm gradually stretched the line to fit the principal curve. The power of the length penalty can be shown by looking at the bottom part of the digit 3. Gradually, from iteration 20 to iteration 60, the bottom part of the 3 becomes straight with the iterations via the length penalty.

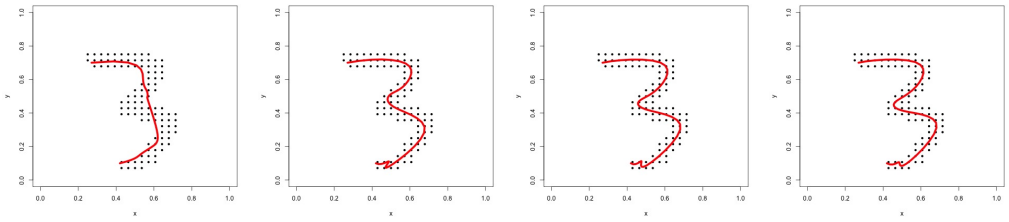


(a) $\lambda = 0.001$ (b) $\lambda = 0.01$ (c) $\lambda = 0.1$

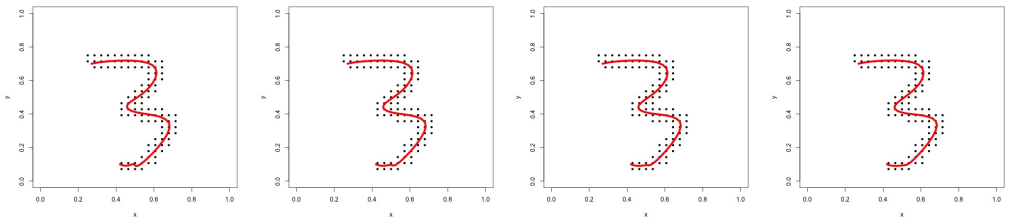


(d) $\lambda = 1$ (e) $\lambda = 10$ (f) $\lambda = 100$

Fig 5: The performance of length penalties



(a) (b) (c) (d)

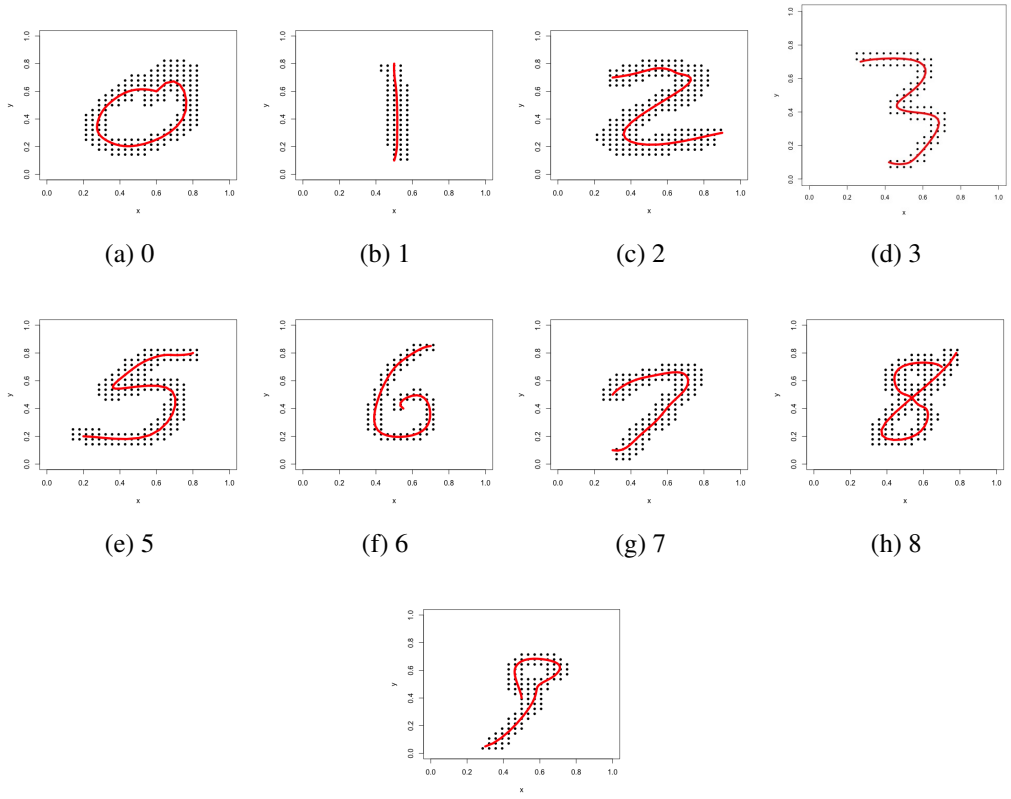


(e) (f) (g) (h)

Fig 6: Fitting procedure of the length penalized principal curve, (a) 5 interations, (b) 10 iteratinos, (c) 15 iterations, (d) 20 iterations, (e) 25 iterations, (f) 30 iterations, (g) 35 iterations, (h) 40 iterations.

Next, we study how the length penalized principal curve performs with different numbers. The result is shown in the Figure 7. The algorithm performs remarkably well. However, several caveats need to be raised. First, number 4 is not included in the analysis, because normally, one writes it in two steps, which is beyond the scope of our discussion here. Second, considering the number 0, it can be seen that the length penalized

principal curve also works well when the curve's start and end connects with each other. Thrid, surprisingly, for the number 8, the length penalized principal curve writes it with a loop. In fact, in Figure 8, it can be seen that whether the number 8 self-loops can be tuned by varying the length-penalty. A large length penalty eliminates it, while a small one creates the loop.

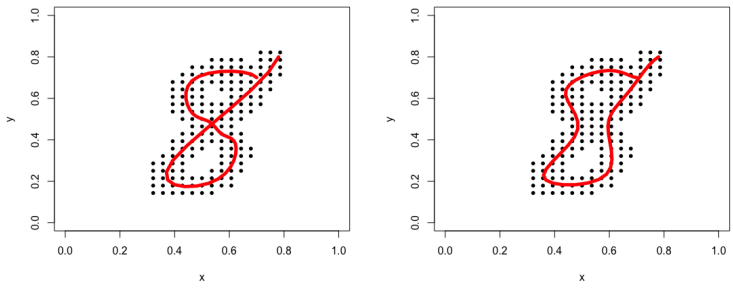


(a) 0 (b) 1 (c) 2 (d) 3

(e) 5 (f) 6 (g) 7 (h) 8

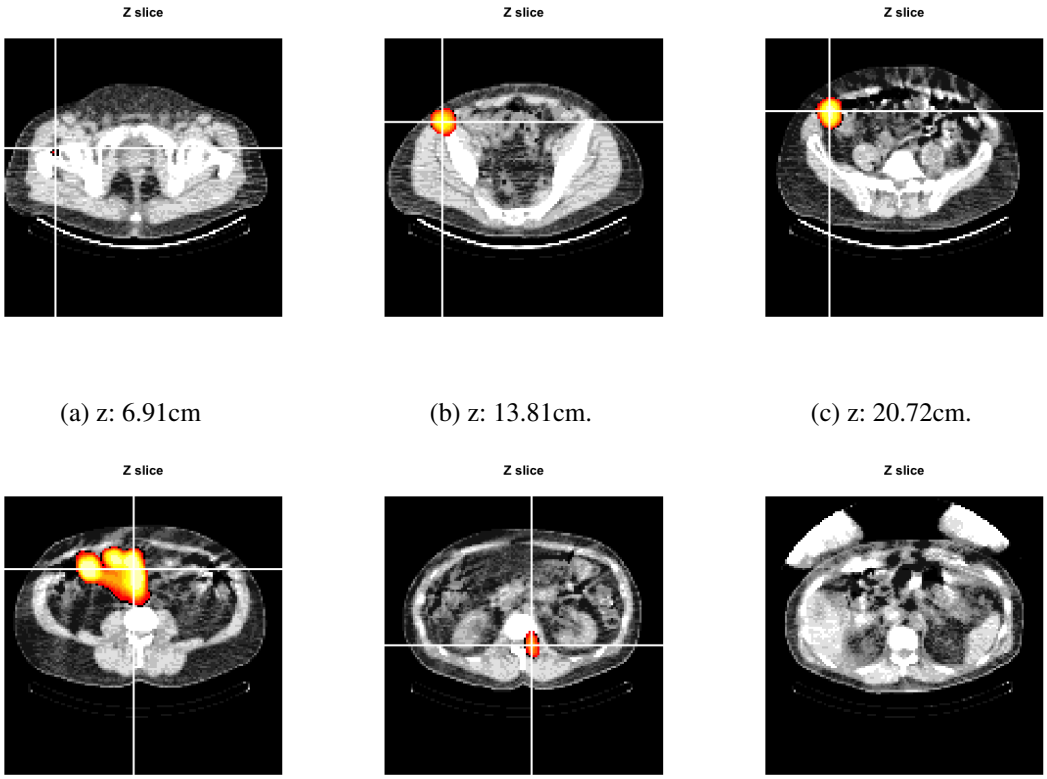
(i) 9

Fig 7: Length Penalized Principal Curve applied to different handwritten digits.



(a) $\lambda = 0.1$ (b) $\lambda = 0.2$

Fig 8: Fitted curves demonstrating that the length penalty controls the writing style of the digit 8, by either crossing or not at the midpoint.



(a) z: 6.91cm (b) z: 13.81cm. (c) z: 20.72cm.

(d) z: 27.63cm. (e) z : 34.53cm. (f) z: 41.44cm.

Fig 9: Coregistered axial slices of CT and SPECT images. In these images the SPECT thresheld was set at 10.

6. Algorithm applied to the colon image dataset. The probabilistic length penalized principal curve was also applied to the SPECT colon image dataset. The SPECT dataset was processed by reconstruction, filtering, thresholding and sampling. In the thresholding stage, the threshold for the image was set at 10, with the number of knots at 200 and degrees of freedom at 10; the length penalty was set to 60. It is worth noticing that, in contrast to the traditional principal curve algorithm, where the total number of knots needs to be tuned manually, with the length penalty, the total number of knots for our principal curve algorithm can be set high, letting the length penalty control curve complexity. The starting and ending points were selected by comparison with bone structure from a co-registered X-ray CT. Cross sectional overlays between the thresheld SPECT and CT images are shown in Figure 9. In Figure 10, we visualize a 3D rendered version of the skelatal CT, length penalized probabilistic fitted curve (shown in red) and the SPECT image data points exceeding the threshold (shown in blue).

In comparison, similar to the experiment conducted in Section 3 where no penalty was applied, the tranditional principal curve algorithm fit to the image is not adequate. In the region where the point density is high, the curve tended to have excessive complexity and curvature (left panel of Figure 12). When a large penalty was applied, the principal curve tended to ignore the shape of the image, just connecting the starting and ending points with a straight line (right side of Figure 12).

The blue point represents the most proximal extent (in the direction of the subject's mouth) of the radiotracer in the rectosigmoid colon, while the green point represents a location very near the anus. These two

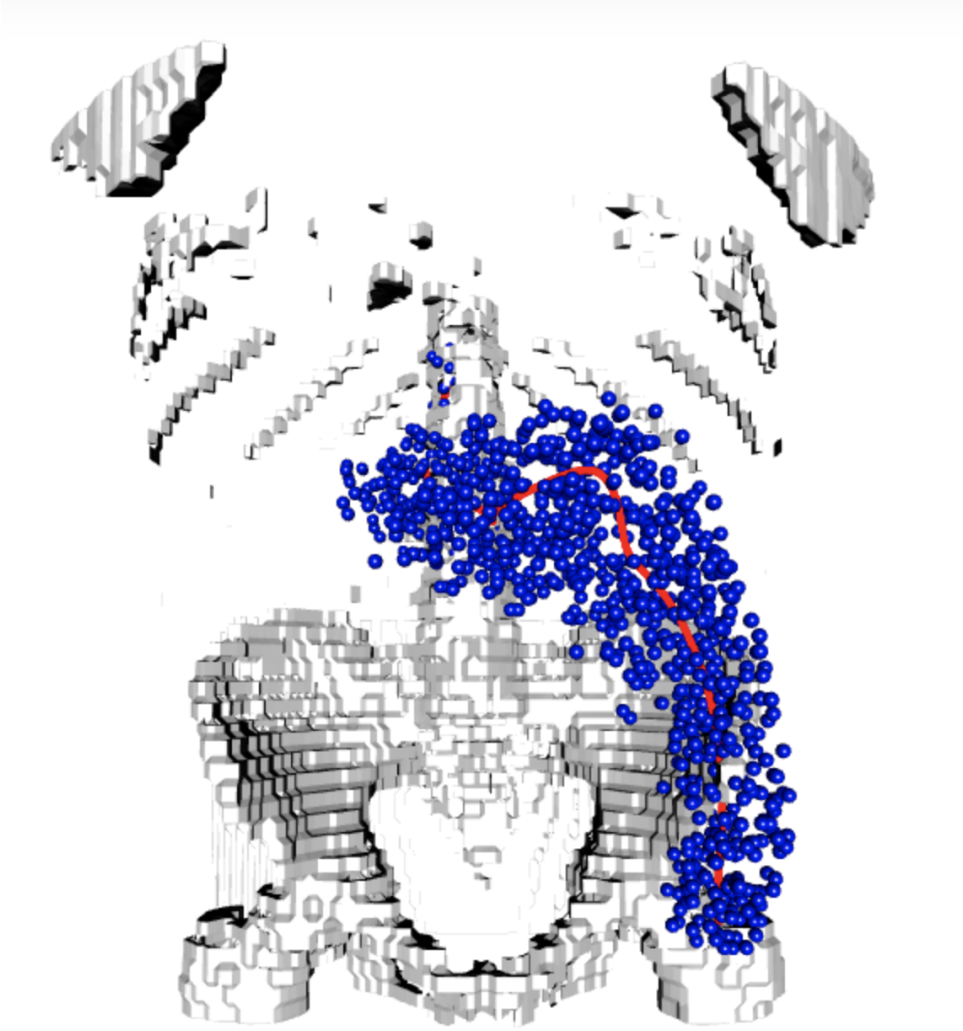
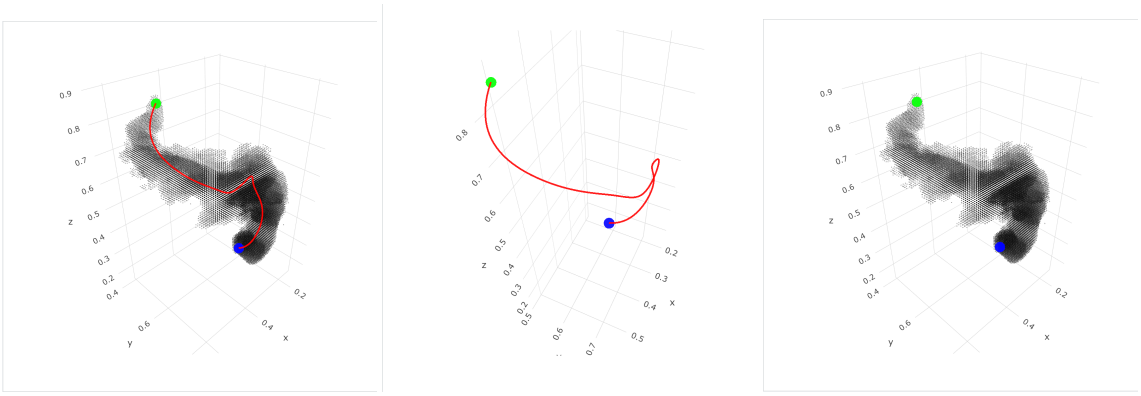


Fig 10: A 3D figure showing the skeleton , curve and data points.

points are pre-specified by visual inspection of the curve. With this information, relevant image quantities of interest can be computed. For instance, the total length of the fitted curve, the average image intensity in neighborhoods along the curve and so on.

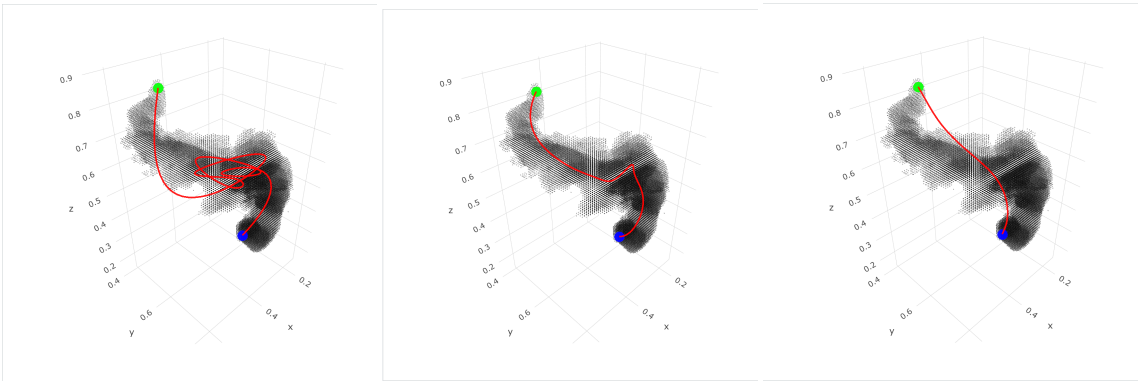
To make the comparison valid, we applied the method used in [14] with cubic splines. In addition, because of numerical issues (the algorithm did not converge without a penalty) a very small ridge penalty was added to fit this curve. In the new method, we applied a higher degree spline (10 degrees) in order to capture the fine structure of the image and control the complexity via the length penalty. Looking at the fitted 3D curve results, more complexity structure can be found, especially in the middle part of the colon, where the old method ignores the irregular structure.

In this setting, the quantity of interest from the SPECT images is a function representing the distribution (concentration) of the enema along the center line. This is accomplished via a moving average of the the image intensities along the curve or more complex projection methods [33]. Figure 13 shows the curves on this subject matching the smoothing degrees of freedom at 50. In the left panel, no length penalty was added



(a) Curve + Point (b) Curve (c) Point

Fig 11: Curve Fitting on the colon image

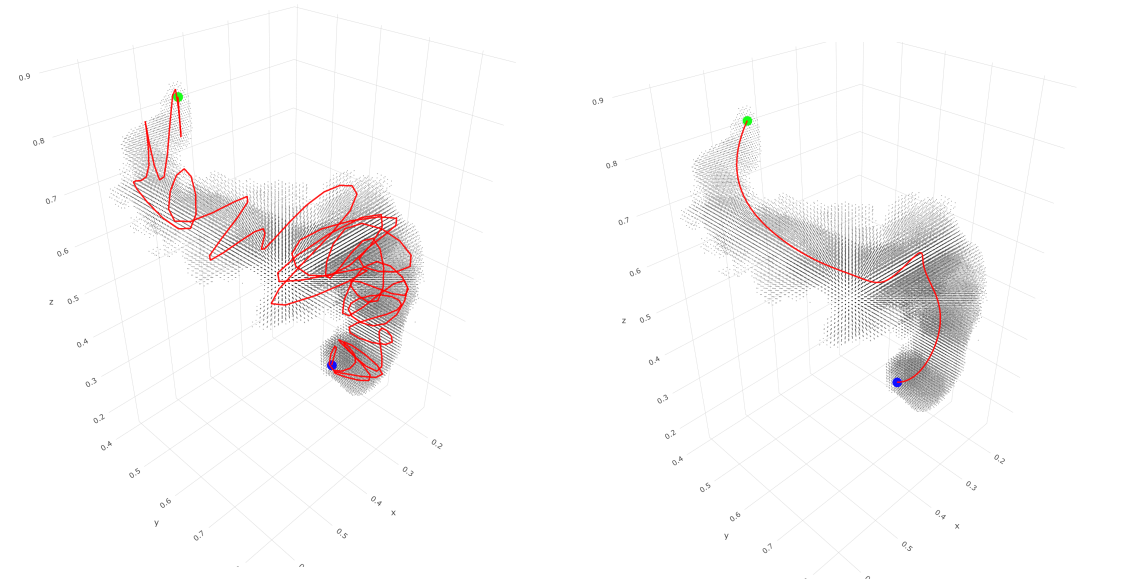


(a) $\lambda = 0.01$ (b) $\lambda = 60$ (c) $\lambda = 10000$

Fig 12: Principal Curve with different penalty value

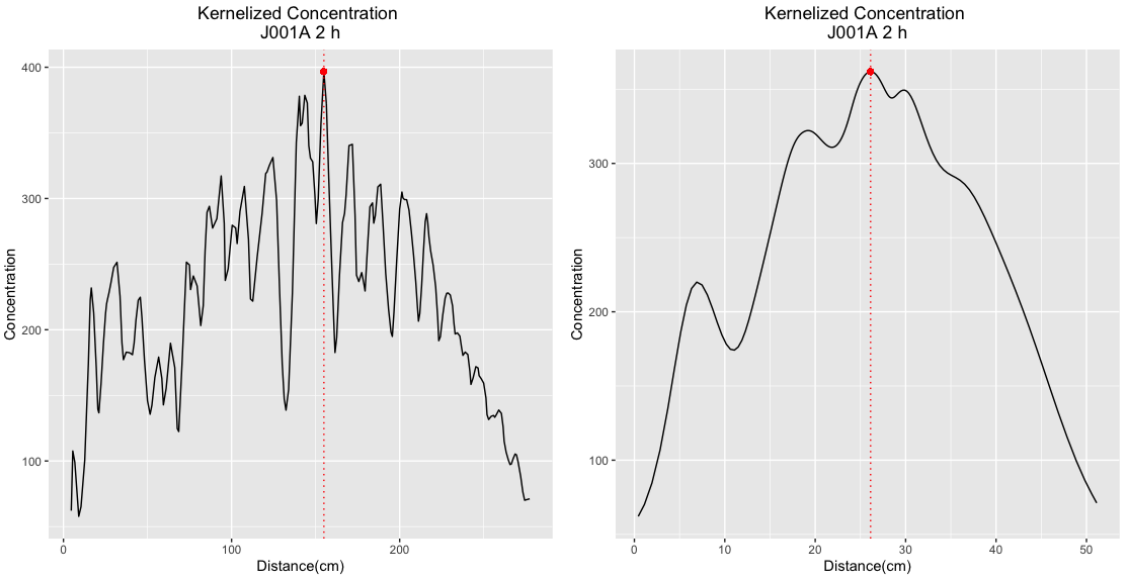
using the method developed in [14], while the right shows the result of the probabilistic length penalized curve. The degrees of freedom was set purposefully high to highlight a lack of robustness in the traditional algorithm. Without the penalty, the measured length of the distribution (a parameter of interest) is on the order of 4 times that of the newer algorithm (notice the different X-axis scales). The fitted curve using the old algorithm utilizes the unnecessary degrees of freedom to loop within the colon distribution to minimize error. Of course, one could combat this with better strategies for the selection of degrees of freedom in the old algorithm. However, the necessary degrees of freedom changes across subjects with a variety of factors including: colon complexity, distribution of points above the threshold and the threshold itself. In contrast, the length penalty is more consistent across subjects, as distributional length is more consistent across subjects.

7. Discussion. In this paper, we proposed a new length penalized probabilistic principal curve algorithm which mainly extends the algorithm proposed in [14]. This algorithm utilizes a probabilistic framework which can be estimated with the EM algorithm and makes use of a length penalty to avoid the self intervening problem commonly encountered in real world applications. Our algorithm is pseudo-Bayesian in the sense of applying a prior and latent variable formulation and viewing it as a a penalized variation of maximum likelihood. While, it is important to emphasize that by relying on only a few core statistical tech-



(a) Old method: fitted curve.

(b) New method: fitted curve.



(c) Old method: distance / concentration curve.

(d) New method: distance / concentration curve.

Fig 13: Comparison of principal curve algorithms. Shown are the fitted 3D curves on SPECT image voxels above the threshold using the older and newer algorithms with smoother degrees of freedom held constant at 50 and the distance / concentration functions associated with the fitted curves. Note the change in X-axis scale on the distance / concentration curves.

niques (principal curves, smoothing, penalization, EM) our algorithm is both easy to implement from scratch and explain, more fully Bayesian solutions may lead to preferable fits. Related work includes [34, 35, 36], which considers fitting curves to the MNIST data and incorporating prior curve information. An interesting direction for our colon application would utilize Bayesian methodology for a collection of curves across individuals with spatially registered CT (skeletal) images. This would allow for more full use of common

colon anatomy in fitting the curve. Further, final usage of the fitted curves requires the knowledge of anatomical lengths (anus to the start of microbicide distribution) that are not available in the image. A Bayesian solution that combines proxies for this distance utilizing imaged bony structures like the coccyx along with uncertainty quantified in a prior would allow for more accurate uncertainty quantification in extrapolated curve components.

With regard to general uncertainty quantification, our strategy in the past has been to use residual bootstrapping. Again, fully Bayesian procedure typically include uncertainty quantification of derived curve quantities as byproducts of MCMC fitting algorithms. However, it is important to emphasize that in both of these solutions, our statistical setting is unusual and unmodeled variability in image intensities may be more of interest for quantifying uncertainty than spatial variation of points around the curve. Currently, this spatial variation is dependent on somewhat arbitrary thresholding steps prior to analysis. An alternative strategy would use image intensities as weights in the likelihood and omit thresholding (or only apply mild thresholding for computational simplicity).

In our application, utilizing the length penalty has solved an issue of requiring large amounts of user input in the form of constrained interior points. The length penalty appears to be an essential component to obtain fits that mirror intuition. However, all of our approaches used non-adaptive smoothing (see [37, 37]). It possible that procedures adaptive with respect to the latent principal parameters will be able to fit more complex features. In addition, we used a smoother that was independent across coordinates. A more complex form of spatial dependence may result in better parameter estimation if joint information can be exploited.

References.

[1] Elizabeth G McFarland, Ge Wang, James A Brink, Dennis M Balfe, Jay P Heiken, and Michael W Vannier. Spiral computed tomographic colonography: determination of the central axis and digital unraveling of the colon. *Academic radiology*, 4(5):367–373, 1997.

[2] Yaseen Samara, Martin Fiebich, Abraham H Dachman, Kunio Doi, and Kenneth R Hoffmann. Automated centerline tracking of the human colon. In *Medical Imaging 1998: Image Processing*, volume 3338, pages 740–746. International Society for Optics and Photonics, 1998.

[3] Lichan Hong, Shigeru Muraki, Arie Kaufman, Dirk Bartz, and Taosong He. Virtual voyage: Interactive navigation in the human colon. In *SIGGRAPH*, volume 97, pages 27–34, 1997.

[4] Rui CH Chiou, Arie E Kaufman, Zhengrong Liang, Lichan Hong, and Miranda Achniotou. Interactive path planning for virtual endoscopy [colon ct]. In *1998 IEEE Nuclear Science Symposium Conference Record. 1998 IEEE Nuclear Science Symposium and Medical Imaging Conference (Cat. No. 98CH36255)*, volume 3, pages 2069–2072. IEEE, 1998.

[5] Thomas Deschamps and Laurent D Cohen. Fast extraction of minimal paths in 3D images and applications to virtual endoscopy. *Medical image analysis*, 5(4):281–299, 2001.

[6] Elaine Cohen, Richard F Riesenfeld, and Gershon Elber. *Geometric modeling with splines: an introduction*. AK Peters/CRC Press, 2001.

[7] Laurent D Cohen and Ron Kimmel. Global minimum for active contour models: A minimal path approach. *International journal of computer vision*, 24(1):57–78, 1997.

[8] Parag Chaudhuri, Rohit Khandekar, Deepak Sethi, and Prem Kalra. An efficient central path algorithm for virtual navigation. In *Proceedings Computer Graphics International, 2004.*, pages 188–195. IEEE, 2004.

[9] I Bitter, M Sato, MA Bender, A Kaufman, M Wan, and MR Wax. Automatic, accurate and robust colon centerline algorithm. In *Radiology*, volume 217, pages 370–370. RADIOLOGICAL SOC NORTH AMER 20TH AND NORTHAMPTON STS, EASTON, PA 18042 USA, 2000.

[10] Edsger W Dijkstra. A note on two problems in connexion with graphs. *Numerische mathematik*, 1(1):269–271, 1959.

[11] Trevor Hastie and Werner Stuetzle. Principal curves. *Journal of the American Statistical Association*, 84(406):502–516, 1989.

[12] Robert Tibshirani. Principal curves revisited. *Statistics and computing*, 2(4):183–190, 1992.

[13] Derek Stanford and Adrian E Raftery. Principal curve clustering with noise. Technical report, Citeseer, 1997.

[14] Brian S Caffo, Ciprian M Crainiceanu, Lijuan Deng, and Craig W Hendrix. A case study in pharmacologic colon imaging using principal curves in single-photon emission computed tomography. *Journal of the American Statistical Association*, 103(484):1470–1480, 2008.

[15] Horst Friedsam and W Oren. The application of the principal curve analysis technique to smooth beam lines. *eConf*, 8907312(SLAC-PUB-11408):011, 1989.

[16] Balázs Kégl and Adam Krzyzak. Piecewise linear skeletonization using principal curves. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(1):59–74, 2002.

[17] Klaus Reinhard and Mahesan Niranjan. Parametric subspace modeling of speech transitions. *Speech Communication*, 27(1):19–42, 1999.
[18] Chris Brunsdon. Path estimation from GPS tracks. In *Proceedings of the 9th International Conference on GeoComputation*. National Centre for Geocomputation, Maynooth University., 2007.
[19] Jeffrey D Banfield and Adrian E Raftery. Ice floe identification in satellite images using mathematical morphology and clustering about principal curves. *Journal of the American Statistical Association*, 87(417):7–16, 1992.
[20] Jochen Einbeck, Gerhard Tutz, and Ludger Evers. Exploring multivariate data structures with local principal curves. In *ClassificationThe Ubiquitous Challenge*, pages 256–263. Springer, 2005.
[21] Glenn Death. Principal curves: a new technique for indirect and direct gradient analysis. *Ecology*, 80(7):2237–2253, 1999.
[22] Wilbur CK Wong and Albert CS Chung. Principal curves to extract vessels in 3D angiograms. In *2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pages 1–8. IEEE, 2008.
[23] Francisco J Leyva, Rahul P Bakshi, Edward J Fuchs, Liye Li, Brian S Caffo, Arthur J Goldsmith, Ana Ventuneac, Alex Carballo-Diéguez, Yong Du, Jeffrey P Leal, et al. Isoosmolar enemas demonstrate preferential gastrointestinal distribution, safety, and acceptability compared with hyperosmolar and hypoosmolar enemas as a potential delivery vehicle for rectal microbicides. *AIDS research and human retroviruses*, 29(11):1487–1495, 2013.
[24] Nicolette A Louissaint, Sridhar Nimmagadda, Edward J Fuchs, Rahul P Bakshi, Ying CAO, Linda A Lee, Jeff Goldsmith, Brian S Caffo, Yong Du, Karen E King, et al. Distribution of cell-free and cell-associated HIV surrogates in the colon following simulated receptive anal intercourse in men who have sex with men. *Journal of acquired immune deficiency syndromes (1999)*, 59(1):10, 2012.
[25] Francisco Leyva, Edward J Fuchs, Rahul Bakshi, Alex Carballo-Dieguez, Ana Ventuneac, Chen Yue, Brian Caffo, Yong Du, Michael Torbenson, Liye Li, et al. Simultaneous evaluation of safety, acceptability, pericoital kinetics, and ex vivo pharmacodynamics comparing four rectal microbicide vehicle candidates. *AIDS research and human retroviruses*, 31(11):1089–1097, 2015.
[26] Ethel Weld, Edward Fuchs, Mark Marzinke, Peter Anton, Ken Ho, Rahul Bakshi, Jarrett Engstrom, Julie Elliott, Lisa Rohan, Cindy Jacobson, et al. Tenofovir douche for prep: On-demand, behaviorally-congruent douche rapidly achieves colon tissue concentration targets (dream 01 study). In *AIDS RESEARCH AND HUMAN RETROVIRUSES*, volume 34, pages 71–71. MARY ANN LIEBERT, INC 140 HUGUENOT STREET, 3RD FL, NEW ROCHELLE, NY 10801 USA, 2018.
[27] Trevor Hastie, Robert Tibshirani, Jerome Friedman, and James Franklin. The elements of statistical learning: data mining, inference and prediction. *The Mathematical Intelligencer*, 27(2):83–85, 2005.
[28] Paul HC Eilers and Brian D Marx. Flexible smoothing with B-splines and penalties. *Statistical science*, pages 89–102, 1996.
[29] David Ruppert, Matt P Wand, and Raymond J Carroll. *Semiparametric regression*. Number 12. Cambridge university press, 2003.
[30] Xihong Lin and Daowen Zhang. Inference in generalized additive mixed modelsby using smoothing splines. *Journal of the royal statistical society: Series b (statistical methodology)*, 61(2):381–400, 1999.
[31] Yann LeCun and Corinna Cortes. MNIST handwritten digit database. 2010.
[32] Arthur P Dempster, Nan M Laird, and Donald B Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)*, 39(1):1–22, 1977.
[33] Jeff Goldsmith, Brian Caffo, Ciprian Crainiceanu, Daniel Reich, Yong Du, and Craig Hendrix. Nonlinear tube-fitting for the analysis of anatomical and functional structures. *The annals of applied statistics*, 5(1):337, 2011.
[34] Minhua Chen, Jorge Silva, John Paisley, Chunping Wang, David Dunson, and Lawrence Carin. Compressive sensing on manifolds using a nonparametric mixture of factor analyzers: Algorithm and performance bounds. *IEEE Transactions on Signal Processing*, 58(12):6140–6155, 2010.
[35] Brian Neelon and David B Dunson. Bayesian isotonic regression and trend analysis. *Biometrics*, 60(2):398–406, 2004.
[36] Ye Wang and David B Dunson. Probabilistic curve learning: Coulomb repulsion and the electrostatic Gaussian process. In *Advances in Neural Information Processing Systems*, pages 1738–1746, 2015.
[37] Tatyana Krivobokova, Ciprian M Crainiceanu, and Göran Kauermann. Fast adaptive penalized splines. *Journal of Computational and Graphical Statistics*, 17(1):1–20, 2008.

SUPPLEMENTARY MATERIAL

Supplement A: R package
(https://github.com/CHuanSite/ppclp). A R package of the algorithm is developed for this paper. This package includes both 2D and 3D curve fitting.

Supplement B: Shiny App
(https://ppclp.shinyapps.io/ppclpshiny/). A shiny app for the algorithm in this paper.

Supplement C: Procedure to implement Penalized EM

().

E-Step:

The full data likelihood is

$$L = \prod_{i=1}^{n} \prod_{k=1}^{K} [p(\mathbf{x}_i|t_k)\pi_k]^{1_{\lambda_i=t_k}}$$

Take logarithm of L, we have

$$\log L = \sum_{i=1}^{n} \sum_{k=1}^{K} 1_{\lambda_i=t_k} [\log \pi_k + \log p(\mathbf{x}_i|t_k)]$$

Take expectation of λ_i conditional on X to $\log L$

$$E_{\lambda|X}[\log L] = \sum_{i=1}^{n} \sum_{k=1}^{K} \theta_{ik}^{(t-1)} [\log \pi_k + \log p(\mathbf{x}_i|t_k)]$$

where

$$\theta_{ik}^{(t-1)} = P(\lambda_i = t_k|\mathbf{X}_i) = \frac{p^{(t-1)}(\boldsymbol{x}_i|t_k)\pi_k^{(t-1)}}{\sum_k p^{(t-1)}(\boldsymbol{x}_i|t_k)\pi_k^{(t-1)}}$$

M-Step:

$$\hat{\pi}^t = \frac{\sum_{i=1}^{N} \theta_{ik}^{(t-1)}}{N}$$

$$\hat{\sigma}_x^{2(t)} = \frac{1}{N} \sum_{i=1}^{N} \sum_{k=1}^{K} \theta_{ik}^{(t-1)} \{(x_i - \boldsymbol{B}^\top(t_k)\hat{\boldsymbol{\beta}}_x^{(t)})^2\}$$

$$\hat{\sigma}_y^{2(t)} = \frac{1}{N} \sum_{i=1}^{N} \sum_{k=1}^{K} \theta_{ik}^{(t-1)} \{(y_i - \boldsymbol{B}^\top(t_k)\hat{\boldsymbol{\beta}}_y^{(t)})^2\}$$

$$\hat{\sigma}_z^{2(t)} = \frac{1}{N} \sum_{i=1}^{N} \sum_{k=1}^{K} \theta_{ik}^{(t-1)} \{(z_i - \boldsymbol{B}^\top(t_k)\hat{\boldsymbol{\beta}}_z^{(t)})^2\}$$

$$\hat{\boldsymbol{\beta}}_x^{(t)} = (\boldsymbol{B}^\top \boldsymbol{W}_1^{(t-1)} \boldsymbol{B})^{-1} \boldsymbol{B}^\top \boldsymbol{W}_2^{(t-1)} x$$

$$\hat{\boldsymbol{\beta}}_y^{(t)} = (\boldsymbol{B}^\top \boldsymbol{W}_1^{(t-1)} \boldsymbol{B})^{-1} \boldsymbol{B}^\top \boldsymbol{W}_2^{(t-1)} y$$

$$\hat{\boldsymbol{\beta}}_z^{(t)} = (\boldsymbol{B}^\top \boldsymbol{W}_1^{(t-1)} \boldsymbol{B})^{-1} \boldsymbol{B}^\top \boldsymbol{W}_2^{(t-1)} z$$

where W_1, W_2 are both weighted probability matrix used in the EM algorithm, where

$$(\boldsymbol{W}_1^{(t-1)})_{ks} = \begin{cases} \sum_{i=1}^{N} \theta_{ik}^{(t-1)} & k = s \\ 0 & \text{else} \end{cases}$$

$$(\boldsymbol{W}_2^{(t-1)})_{ij} = \theta_{ji}^{(t-1)}$$