Estimations of Principal Curves

Dongwoon Lee
Department of Computer Science
University of Toronto
Toronto, Ontario
dwlee@dgp.toronto.edu

Abstract

Principal curves are smooth curves that minimize the average squared orthogonal distance to each point in a data set. Fitting a principal curve is a maximum-likelihood technique for nonlinear regression in the presence of Gaussian noise on both x and y. We choose two definitions of principal curves in the literature and then present experimental results to discuss over them.

1 Introduction

A principal curves is a smooth curve passing through the 'middle' of a distribution or data cloud, and is a generalization of linear principal components. In other words, a principal curves is a set of points which represent well the mean of data densities.

Several definitions of principal curves have been proposed in the literature. Hastie and Stuetzle [1] (hereafter HS) proposed the earliest definitions which is based on 'self-consistency', i.e. the curve should coincide at each position with the expected value of the data projecting to that position. Tibshirani [2] (hereafter EM-based) presented a more probabilistic approach, of which the principal curve is defined as curves minimizing a penalized log-likelihood measure with Gaussian mixtures and generalized EM algorithms. Kegl et al.[3] and Verbeek et al.[4] proposed another approaches to define principal curves as continuous curves of a given length which minimize the expected squared distance between the curve and points of the space randomly chosen according to given distribution. They introduced incremental method by fitting local models without any topological constraints and then increasing complexity.

We choose two earliest definitions of principal curves mentioned above; one is for HS [1] and the other one for EM-based [2]. In the next sections, we present each of definitions and algorithms in detail. Then we give experimental results so that we compare and discuss over them.

2 Principal Curves

We first give a brief introduction to one-dimensional curves, and then present each of principal curves definitions in probability distributions, algorithms for finding curves. In the last section, we briefly mention the regularization method, 'cubic smooth splines' we implement in our project.

2.1 One-Dimensional Curves

A one-dimensional curve in p-dimensional space is a vector $f(\lambda)$ of p functions of a single variable λ. These functions are the coordinate functions, and λ provides an ordering along the curve. If the coordinate functions are smooth, then f is to be definition a smooth curve. There is a natural parameterization for curves in terms of the arc length. The arc length of a curve f from λ_0 to λ_1 is given by $l = \int_{\lambda_0}^{\lambda_1} \|f'(z)\| dz$. If the curve is unit parameterized, i.e. $\|f'(z)\| \equiv 1$, $l = \lambda_1 - \lambda_0$.

2.2 HS Principal Curves

2.2.1 Definition

Consider $p-$dimensional random vector $X = (X_1, \ldots, X_p)$ with finite second moments. Let f denote a smooth (C^∞) unit-speed curve in R^p parameterized over a closed interval, that dose not intersect itself($\lambda_1 \neq \lambda_2 \Rightarrow f(\lambda_1) \neq f(\lambda_2)$) and has finite length inside any finite ball in R^p. The principal curve f has the property that the expected value of the squared Euclidean distance from X to the curve f. We define the projection index $\lambda_f : R^p \rightarrow R^1$ as

$$\lambda_f(x) = \sup_{\lambda} \{\lambda : \|x - f(\lambda)\| = \inf_{\mu} \|x - f(\mu)\|\} \tag{1}$$

The projection index $\lambda_f(x)$ of x is the value of λ for which $\lambda_f(x)$ is closest to x. If there are several values, we pick the largest one.

The curve f is self-consistent if every single point $f(\lambda)$ along the curve f coincides the conditional expectation value of randomly distributed points projected to $f(\lambda)$.

$$f(\lambda) = E(X \mid \lambda_f(X) = \lambda) \tag{2}$$

2.2.2 Algorithm

By analogy to linear principal component analysis, we are finding smooth curves corresponding to local minima of the distance function. We start with principal component line, which is also self-consistent. Our estimation procedure consists of two steps; projection step and expectation step. Every sample points are projected on the curve (projection step) and then conditional expected values are computed with those projected points set(expectation step). By enforcing self-consistency property, those expected values should be equal to projected points. If they do not coincide expected values, the resulting curve should be changed according to new set of expected points. As both of projection and expectation step iteratively reduce expected squared distance, our procedure get converged to the principal curve.

Initialization : $f^{(0)}(\lambda) = E(X) + a\lambda$, where a is the first linear principal component of distributed density h. Set $f^{(0)}(x) = \lambda_{f^{(0)}}(x)$.

Repeat until the change in $D^{(0)}(h, f^{(j)})$ below some threshold,

$\lambda_x^{(j)} = \lambda_{f^{(j)}}(x) \quad \forall x \in h$: (projection step)

$f^{(j)}(\lambda_x^{(j)}) = E(X \mid \lambda_{f^{(j)}}(x) = \lambda_x^{(j)})$: (expectation step)

$D^2(h, f^{(j)}) = E_{\lambda^{(j)}} E[\|X - f(\lambda^{(j)}(X))\| \mid f(\lambda^{(j)}(X))]$

2.2.3 Principal Curves for Dataset

A curve $f(\lambda)$ is represented by at most n tuples $(\lambda_i, f_i(\lambda_i))$, which can be regarded as knots of interpolated cubic curve, and join up curve segments in increasing order of λ_i. We compute λ_i by using arc-length parameterization, and always sort up and normalize them in [0,1]. Arc-length is computed by measuring polygonal line length along the curve, which is discrete version of the unit-speed parameterization.

At projection step, we define d_{ik} as the distance between x_i and its closest point on the line segment joining ($f^{(j)}(\lambda_i), f^{(j)}(\lambda_{i+1})$). After every d_{ik} and corresponding λ_i are computed, we replace λ_i by arc-length of $f_1^{(j)}$ to $f_i^{(j)}$.

Expectation step is to estimate $f^{(j+1)}(\lambda) = E(X \mid \lambda_{f^{(j)}}(X) = \lambda)$. Unfortunately, since our data density is finite and restricted in discrete space, there is generally only one such observation for this step. Thus, as the resulting curve is supposed to visit every sample point, our estimation procedure reaches global minima too fast, $d_{ik} \to 0$ and gets uninteresting results. In order to avoid this problem, we estimate conditional expectation at λ_i by averaging all of the observations x_k in the sample for which λ_k is close enough to λ_i.As long as we include more observations into same neighborhood, the underlying conditional expectation is smoother and variance decreases. There are many local smoothing methods to help avoid global minimum problem.

We rely on 'Locally Weighted Running-Line Smoother', which is one of locally averaging methods [6]. We first specify spherical span to each λ_i, where any λ_j should be considered as its neighbor if they fall in. They get weighted according to the distance between λ_i and λ_j, $w_{ij} = (1 - |(\lambda_j - \lambda_i)/h_i|^3)^3$. Derived weights smoothly die to 0 within the neighborhood.

2.3 EM-based Principal Curves

2.3.1 Definition

Let $X = (X_1, \ldots, X_p)$ be a random vector with density $g_X(x)$. In order to define principal curves, we assume that there is a latent variable S generated according to $g_S(s)$, and sample X is generated from a conditional distribution $g_{X|s}$ with mean $f(S)$, a point on a curve in R^p, with $X_1, \ldots, X_p$ conditionally independent given s. Hence, we define a principal curve of g_X to be a triplet $\{g_S, g_{X|s}, f\}$ satisfying the following conditions:

a. $g_S(s)$ and $g_{X|S}(x \mid s)$ are consistent with $g_X(x)$, that is, $g_X(x) = \int g_{X|s}(x \mid s) g_S(s) ds$

b. $X_1, \ldots, X_p$ are conditionally independent given s.

c. $f(S)$ is a curve in R^p parameterized over a closed interval in R^1, satisfying $f(S) = E(X \mid S = s)$

This new definition does not agree with HS property, which is that the X values generated from S are exactly the X values on the projection line orthogonal to $f(S)$ at s. Instead, it helps principal curves overcome model bias problem HS suffered [1]. Except cases involving this problem, it goes well with HS definition.

2.3.2 Algorithm

Estimation procedure follows generalized EM algorithm[5]. Here, since principal curve knots roles latent variables S, curve can be updated according to EM iterations. Instead of least square error functions given by HS, maximum log-likelihood function is considered to EM step, which works well with EM based curve model given by previous section. Initial S positions are specified by HS algorithm's first step and then iteratively updated by alternating expectation and maximization step for this likelihood function. In the subsequent section, more details are presented for log-likelihood function.

Initialization : $f^{(0)}(s) = E(X) + ds$, where d is the first linear principal component of distributed density h. Set $v_k^{(0)} = 1/n$, and S = $a_1^{(0)},\ldots,a_n^{(0)}$

Repeat until the change in log-likelihood below some threshold,

Compute weights, $w_{ik}^{(j+1)} = g_{X|S}(x_i \mid a_k^{(j)}, \theta) v_k^{(j)} / g_X(x_i)$ (with $\sum_{k=1}^{n} w_{ik}^{(j+1)} = 1$.)

$$f^{(j+1)}(a_k) = \sum_{i=1}^{n} w_{ik}^{(j+1)} x_i / \sum_{i=1}^{n} w_{ik}^{(j+1)}$$

$$\sigma^{2(j+1)}(a_k) = \sum_{i=1}^{n} w_{ik}^{(j+1)} (x_i - f^{(j+1)}(a_k))^2 / \sum_{i=1}^{n} w_{ik}^{(j+1)}$$

$$v_k^{(j+1)}(a_k) = 1/n \sum_{i=1}^{n} w_{ik}^{(j+1)}$$

Update curve according to new set of parameters.

2.3.3 Principal curves for Dataset

Suppose we have observations X and hidden variables S, the principal curve is formed via the following model: $s_i \sim g_S(s)$; $X_i \sim g_{X|s_i}$; $f(s) = E(X \mid s)$ (with x_i is conditionally independent to given s_i). Hence, instead of least square error function used by HS, the maximum likelihood estimation of $\theta = (f(s), \sum s)$ is considered for EM algorithm. This maximum likelihood estimation has the form of a mixture: $l(\theta) = \sum_{1}^{n} \log \int g_{X|S}(x_i \mid \theta) g_S(s) ds$.

"E-step" start with $Q(\theta \mid \theta^0) = E\{l_0(\theta) \mid x, \theta^0\}$ and then "M-step" maximizes $Q(\theta \mid \theta^0)$ over θ to give θ^1 and the process is iterated until convergence.

$$Q(\theta \mid \theta^0) = \sum_{i=1}^{n} \sum_{k=1}^{n} w_{ik} \log g_{X|S}(x_i \mid \theta(a_k)) + \sum_{i=1}^{n} \sum_{k=1}^{n} w_{ik} \log v_k \tag{3}$$

$$w_{ik} \sim g_{Y|S}(x_i \mid a_k, \theta) v_k / g_X(x_i) \text{ (by Bayestheorem)} \tag{4}$$

Those quantities are computed from the Gaussian form and related operations:

$$g_X(x_h) = \sum_{k=1}^{n} g_{X|S}(x_h \mid a_k, \theta^0) v_k \,,\; g_{X|S}(x_i \mid \theta(s)) = \prod_{j=1}^{p} \phi_{f_j(s),\sigma_j(s)}(x_{ij})$$

where $\phi_{f_j(a_k),\sigma_j(a_k)}(x\ ) = \sigma_j(a_k)^{-1}(2\pi)^{-1/2} \exp[-(x - f_j(a_k))^2 / 2\sigma_j^2(a_k)]$. (5)

Then, M-step gives

$$\hat{f}(a_k) = \sum_{i=1}^{n} w_{ik} x_i / \sum_{i=1}^{n} w_{ik}$$

$$\hat{\sigma}^2(a_k) = \sum_{i=1}^{n} w_{ik} (x_i - \hat{f}(a_k))^2 / \sum_{i=1}^{n} w_{ik}$$

$$\hat{v}_k(a_k) = 1/n \sum_{i=1}^{n} w_{ik} \tag{6}$$

The weights are the relative probability under the current model that $s = a_k$ gave rise to x_{ij}. If σ_j s are equal, the weight is a function of the Euclidean distance from x to $f(a_k)$. This log-likelihood function can get to a global maximum of $+\infty$, when $\hat{f}(a_k) = x_k, k = 1,2\ldots,n$. $\hat{\sigma}^2(a_k) \to 0$. This problem is exactly same to what HS suffered from. Thus, in order to make EM converge to local maximum, a regularization component is added to the log-likelihood. We seek to maximize the penalized log-likelihood

$$l'(\theta) = l(\theta) - (c_2 - c_1) \sum_{j=1}^{p} \lambda_j \int_{c_1}^{c_2} [f_j''(s)]^2 ds \tag{7}$$

c_2, c_1 are the endpoints of the smallest interval containing the support of $g_S(s)$. Thus corresponding Q function for EM is

$$Q(\theta \mid \theta^0) = \sum_{i=1}^{n} \sum_{k=1}^{n} w_{ik} \log g_{X|S}(x_i \mid \theta(a_k)) + \sum_{i=1}^{n} \sum_{k=1}^{n} w_{ik} \log v_k - (c_2 - c_1) \sum_{j=1}^{p} \lambda_j \int_{c_1}^{c_2} [f_j''(s)]^2 ds \tag{8}$$

Then, the solutions are $\hat{f}_j = (D + (c_2 - c_1)\lambda_j K_j)^{-1} D\{D^{-1}\bar{x}_j\}$ (where matrix K_j is the usual quadratic penalty matrix associated with a cubic smoothing spline[6].

2.4 Principal Curves and Smooth Splines

In the previous sections, there two definitions of principal curves are presented. Since both of their properties have the same problems that object functions get global maximum or minimum, they need some way to regularize them. We choose kernel based smoother for HS algorithm and cubic spline smoother for EM based one. Since both are local based smoothers, we can switch kernel based with cubic spline smoother and vice versa. In this section, we give more details on cubic spline smoother. We find $f(s)$ and $s_i \in [0,1]$ so that

$$D^2(f, S) = \sum_{i=1}^{n} \|x_i - f(s_i)\|^2 + \lambda \int_0^1 \|f''(t)\|^2 dt \tag{9}$$

is minimized over all f. Large values of λ produce smoother curves while smaller values produce more wiggly curves. The penalized least square terms shows that if the minimum exists, it should be cubic spline in each coordinate. Since cubic spline smoother can be computed in $O(n)$, it has an advantage over locally weighted

running line smoother we choose in HS, which requires $O(n^2)$. However, performance looks quite similar. Since the solution to (9) is natural cubic spline with n-2 knots, it can be represented in terms of the unconstrained B-spline basis, $\sum_1^{n+2} \gamma_j B_j(s)$, where γ_j are coefficients and the B_j are the cubic B-spline basis functions. With $B_{ij} = B_j(x_i)$ and $\Omega_{ij} = \int B_i''(t) B_j''(t) dt$, (9) can be rewritten as

$$(x - B\gamma)^T (x - B\gamma) + \lambda \gamma^T \Omega \gamma \tag{10}$$

Setting the derivative with respect to γ equal to zero gives

$$(B^T B + \lambda\Omega)\hat{\gamma} = B^T x \,,\; \hat{\gamma} = (B^T B + \lambda\Omega)^{-1} B^T x \tag{11}$$

More computational and practical approach for this formula is presented by De Boor [7], and our cubic spline smoother is based on this.

3 Experimental Results and Discussion

We implemented those two algorithms with C++ and OpenGL graphics library on PC. Although computational time depends on the model complexity, it is not so important factor for completing whole estimating procedures in practice. We randomly distribute samples according to different kinds of generating functions and on purpose make them corrupted by noises under Gaussian model.

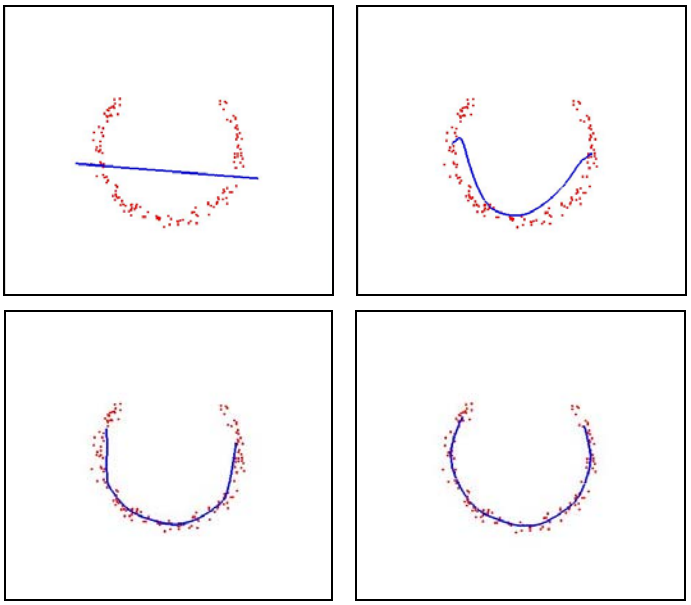

Figure 1: Selected intermediates and final curve of HS Principal Curve Procedures From left to right, top to bottom, n=1,3,4 and 6 iterates.

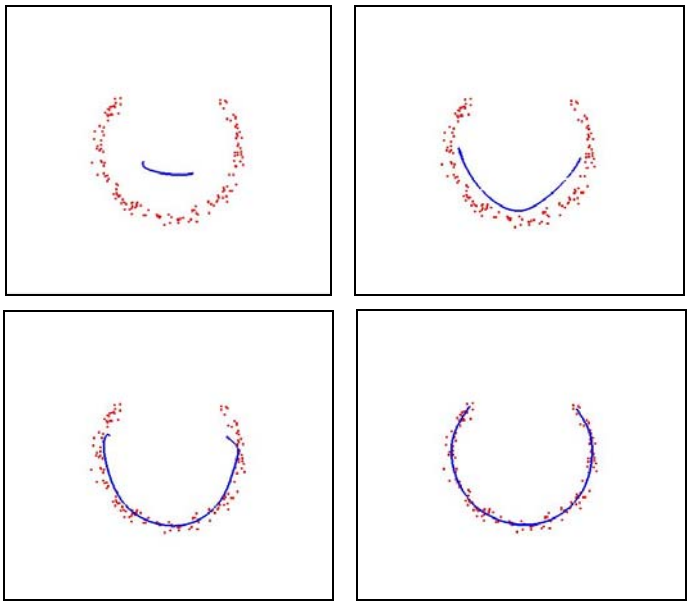

Figure 2: Selected intermediates and final curve of EM based Principal Curve Procedures, From left to right, top to bottom, n=2,5,8 and 12 iterates.

We generate 110 sample points from opened circle in two dimensions:

$(x_1, x_2) = (5\cos(t), 5\sin(t)) + (e_1, e_2)$, where t is uniformly distributed on $[\frac{\pi}{4}, \frac{7\pi}{4}]$ and e_1, e_2 are independent under $N(0,1)$. Since those two algorithms have different convergence criterions, it is difficult to specify same stopping conditions to them and compare with their convergences in numerical aspect. Instead, we can conclude that both of their resulting curves converge to similar shape. However, when using EM-based algorithm, most cases require more iterative steps to finish procedures.

In practice, most cases work well under h=0.2~0.3(kernel span) and λ=0.5~0.6 (cubic smoothing parameters (out of 1)). While wider spans make curves short to converge to average value of all data observations, bigger cubic smoothing parameters give curves more stiffness and inflexibility. However, it is difficult to guess those parameters mechanically without user intervention.

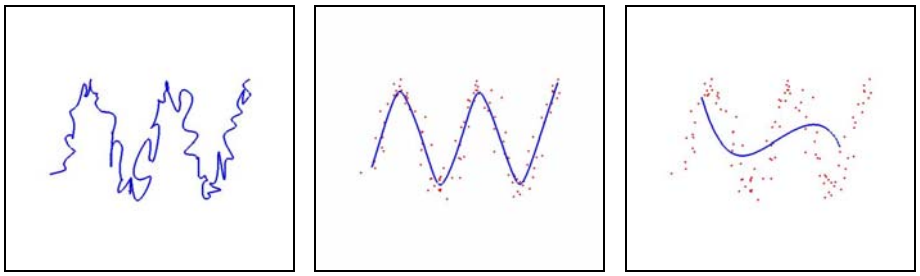

Figure 3: Different results of smoothing parameters (HS algorithm): leftmost curves with h=0.01, λ=0.1, middle one with h=0.25, rightmost with h=0.55 and λ=0.9.

Since those principal curves imply locally topological constraints, they can give out poor performance according to the model complexity. During whole estimation procedures, curves should be connected and knots orders should be an important factor on curve shapes.

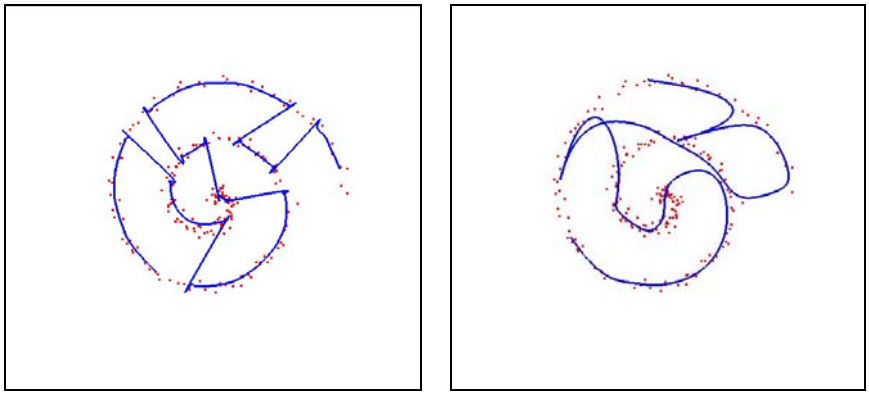

Figure 4: Estimation for spiral distributions: on the left, HS principal curve and on the right, EM based principal curve.

Those challenging problems have already mentioned by many literatures so that many of novel definitions proposed to remedy them [3][4].

3 References

[1] Hastie,T. and Stuetzle, W. Principal curves. *Journal of the American Statistical Association*, 84(406):502-516, 1989.

[2] Tibishirani, R. Principal curves revisted. *Statistics and Computing*, 2:183-190, 1992.

[3] Krzyzak, Linder, T and Zeger, K. Learning and design of principal curves. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(3):281-297, 2000.

[4] Verbeek, J.J., Vlassis, N. and Krose, B. A k-segments algorithm for finding principal curves, *Pattern Recognition Letters*, 23(8), 2002.

[5] Dempster,A.P., Laird, N.M., and Rubin, D.B. Maximum likelihood from incomplete data via the EM algorithm. J.R. *Statist. Soc*. B, 39:1-38, 1997.

[6] Hastie, T. and Tibshirani, R. *Generalized Additive Models*. Chapman and Hall, 1990.

[7] Boor, C. *A Practical Guide to Splines*, 1978.