Available online at www.sciencedirect.com
SCIENCE DIRECT®
Pattern Recognition Letters 27 (2006) 1142–1150
Pattern Recognition Letters
www.elsevier.com/locate/patrec

Automatic parameter selection for a k-segments algorithm for computing principal curves

Haonan Wang *, Thomas C.M. Lee

Department of Statistics, Colorado State University, Fort Collins, CO 80523-1877, USA

Received 24 June 2005; received in revised form 22 December 2005
Available online 8 February 2006

Communicated by W. Pedrycz

Abstract

This paper studies the k-segments algorithm proposed by Verbeek et al. [Verbeek, J.J., Vlassis, N., Krose, B., 2002. A k-segments algorithm for finding principal curves, Pattern Recognition Lett. 23, 1009–1017] for computing principal curves. In particular an automatic method for choosing the "free" parameters in this k-segments algorithm is proposed. Experimental results are provided to demonstrate the performance of this proposed method.

Keywords: Curvilinear feature extraction; k-segments algorithm; Minimum description length principle; Principal curves; Self-consistency; Unsupervised learning

1. Introduction

Hastie and Stuetzle (1989) introduce principal curves as smooth one-dimensional (1D) curves that pass through the "middle" of a set of p-dimensional data points, providing smooth and curvilinear summaries of p-dimensional data. Here a 1D curve in a p-dimensional space is a vector f of p functions indexed by one single variable t. The parameter t is the arc length along the curve.

For any density h in $\mathbb{R}^p$ with finite second moments, the curve f is a principal curve of h if the following *self-consistent criterion* is satisfied for almost every t:

$$E\{X|t_f(X) = t\} = f(t). \quad (1)$$

In the above X is a random vector from h, $t_f(\cdot)$ is the projection index function which maps any value of $X = x$ to the value of t for which $f(t)$ is closest to x. In Hastie and Stuetzle (1989) the practical algorithm for computing a principal curve applies definition (1) iteratively, with the conditional expectation operation approximated by a scatterplot smoothing operation. Alternative definitions for principal curves are given by Tibshirani (1992) and Kegl et al. (2000), while additional algorithms for computing principal curves are developed for examples by Cheng et al. (2004) and Kegl et al. (2000). In addition, the generative topographic mapping of Bishop et al. (1998), the growing cell structures vector quantization technique of Fritzke (1994), and the self-organizing maps vector quantization technique described in Kohonen (1995) can also be applied to compute approximations to principal curves.

However, as pointed out by Verbeek et al. (2002), these algorithms often perform poorly when the data are clustered around highly curved or intersecting structures. In order to solve this issue, Verbeek et al. (2002) propose a so-called k-segments algorithm for finding principal curves. This k-segments algorithm first locate k different line segments in the data set, then these k located segments are linked together to form a *polygonal line*. This polygonal

* Corresponding author. Fax: +1 970 491 7895.
E-mail addresses: wanghn@stat.colostate.edu (H. Wang), tlee@stat.colostate.edu (T.C.M. Lee).

0167-8655/$ - see front matter © 2005 Elsevier B.V. All rights reserved.
doi:10.1016/j.patrec.2005.12.005

line can be used as a first approximation to the principal curve for the data set, and can be further smoothed to obtain a smooth principal curve.

In the practical implementation of this k-segments algorithms, two parameters are required to be chosen. The first parameter is k; that is, the number of line segments. Verbeek et al. (2002) provide a likelihood based method for choosing its value. The second parameter, denoted as λ by Verbeek et al. (2002), is used to determine how the initial segments are linked to form a polygonal line. This parameter λ, roughly speaking, can be treated as a tradeoff parameter for balancing how closely the final resulting principal curve should follow the data and how smooth the curve should be. It is stated in Verbeek et al. (2002) that, especially when the principal curve is self-intersecting, choosing a suitable value for λ is an important issue. Although the idea of using the minimum description length principle of Rissanen (1989) for solving this parameter selection problem has been mentioned by Verbeek et al. (2002), no practical implementation has been reported in Verbeek et al. (2002).

The goal of this paper is to develop an automatic method for simultaneously choosing the values of k and λ. This method is based on the minimum description length principle, and results from numerical experiments demonstrate the good performance of this method. We shall focus on the two-dimensional (2D) setting. Extensions to higher dimensions are straightforward.

The rest of this paper is organized as follows. First a probabilistic model for principal curves and a brief description of the k-segments algorithm of Verbeek et al. (2002) is given in Section 2. The proposed method for choosing k and λ is then presented in Section 3. Results of numerical experiments conducted for evaluating the performance of the proposed method is reported in Section 4. Lastly conclusion is offered in Section 5 while technical details are deferred to Appendix A.

2. Background

2.1. A probabilistic model

Here we describe a probabilistic model for principal curves. Similar models can be found for example in Verbeek et al. (2002) and Delicado and Huerta (2003).

Available is a set $\boldsymbol{x}$ of n spatial points $x_1, \ldots, x_n$ observed in a 2D region. It is assumed that there is one curvilinear feature $\mathscr{F}$ in this 2D region, and that each x_i is independently generated by the following stochastic mechanism. First a point t_i is randomly selected along $\mathscr{F}$. Then x_i is generated at a random distance d_i from t_i in the direction orthogonal to the tangent of $\mathscr{F}$ at t_i, with equal probabilities of being above or below t_i. These random distances d_i's are identically and independently distributed (iid) as $N(0, \sigma^2)$. We assume that the curvilinear feature $\mathscr{F}$ can be well approximated by a principal curve. The goal is, given $\boldsymbol{x}$, to recover $\mathscr{F}$.

2.2. The k-segments algorithm of Verbeek, Vlassis and Kröse

Due to its speed and superior performance, especially in handling self-intersecting features, the k-segments algorithm of Verbeek et al. (2002) rapidly gains its popularity. Here we provide a brief description of this algorithm.

For any given value of k, the k-segments algorithm locates k (disjoint) line segments that attempt to capture the shape characteristics of $\boldsymbol{x}$. Next these k line segments are linked together to form a polygonal line, which can be used as a first approximation to the principal curve for $\boldsymbol{x}$. To determine how these k line segments are linked, the idea of a *Hamiltonian* path is used. For our problem a Hamiltonian path is a linked polygonal line that passes through all n data points $x_1, \ldots, x_n$ exactly once, subject to the constraint that those k line segments are all linked together. The final desired polygonal line is the one that minimizes the following cost function

$$\begin{aligned}\text{cost} = &\{\text{length of the linked polygonal line}\} \\ &+ \lambda\{\text{sum of angles between all pairs} \\ &\quad \text{of adjacent line segments}\}.\end{aligned}$$

In the above the pre-specified parameter λ controls the trade-off between the length and the smoothness of the linked polygonal line.

One can see that the quality of the final principal curve highly depends on the values of k and λ. Verbeek et al. (2002) provide a likelihood based method for determining k, but an automatic method for choosing λ is lacking.

3. The proposed automatic parameter selection method

For a given set of data, applying the k-segments algorithm with different combinations of the parameters (k, λ) would lead to different linked polygonal lines. The aim of this section is to develop an automatic method for choosing the "best" combination of (k, λ). We will make the assumption that the target principal curve that we would like to recover is smooth, but we allow self-intersections in the curve.

Our approach for developing such an automatic parameter selection method is as follows. First we transform the problem of choosing (k, λ) as a statistical model selection problem. As demonstrated below, this is achieved by making various statistical assumptions on the data and also the linked polygonal lines. Then the minimum description length (MDL) principle of Rissanen (1989) is applied to derive a solution to this selection problem. In general, the MDL principle solves a statistical model selection problem by seeking an effective representation, or summary, of the data via the idea of code length minimization. This code length minimization idea has been successfully applied to solve various image and signal processing problems (e.g., see Hansen and Yu, 2000; Lee and Talbot, 1997; Xie et al., 2004), and is expected to perform equally well for the current problem.

For the current problem the MDL principle defines a "best" combination of (k,λ) as the one that enables the best encoding of the data $\boldsymbol{x}$, so that these data can be transmitted (or compressed) in the most economical way. That is, the best (k,λ) combination is the one that produces the shortest description length of $\boldsymbol{x}$. Therefore, in order to apply the MDL principle to tackle this problem, we first need to construct a code length expression which calculates the amount of space (in terms of number bits) that is required to store $\boldsymbol{x}$ for a given (k,λ) combination. Then the best (k,λ) combination is defined as the minimizer of this code length expression. We will use the two-part MDL of Rissanen (1989) to derive such a code length expression.

The idea of the two-part MDL is to first decompose $\boldsymbol{x}$, in a natural manner, into two parts and then encode each part separately. Here this two-part approach suggests that, for each data point x_i we first encode its orthogonal projection $\hat{x}_i$ on the approximated principal curve and then encode the difference vector $\|x_i - \hat{x}_i\|$ between x_i and its projection point $\hat{x}_i$. Now to encode all the projection points $\{\hat{x}_i\}_{i=1}^n$, one could first supply an approximated principal curve (i.e., a polygonal line), and then for each $\hat{x}_i$, the distance, or arc length, that one needs to travel on the polygonal line from one of its endpoints. Denote the polygonal line as P, and the corresponding arc length for $\hat{x}_i$ as t_i. Therefore, a complete knowledge of P and $\{t_i\}_{i=1}^n$ allows a full recovery of all the projection points $\{\hat{x}_i\}_{i=1}^n$. If we use $C(y)$ to denote the code length for the object y, then the code length $C(\boldsymbol{x})$ required to encode $\boldsymbol{x}$ can be expressed as

$$\begin{aligned} C(\boldsymbol{x}) &= C(\{\hat{x}_i\}_{i=1}^n) + C(\{\|x_i - \hat{x}_i\|\}_{i=1}^n) \\ &= C(P) + C(\{t_i\}_{i=1}^n) + C(\{\|x_i - \hat{x}_i\|\}_{i=1}^n) \end{aligned} \quad (2)$$

Now the task is to obtain a computable expression for $C(\boldsymbol{x})$ so that the best combination of (k,λ) can be obtained as its minimizer.

3.1. Code length calculation for $C(P)$

We need some additional notation to proceed (see Fig. 1). A reconnected polygonal line P is composed of a series of line segments and links; i.e., the detected line segments are linked together by the links to form the polygonal line P. Denote the length of the ith line segment as S_i and the length of the jth link as L_j, where $i = 1,\ldots,k$ and $j = 1,\ldots,k-1$. Let z and m respectively be the position of the free endpoint and the slope of the first segment. Also let α_i and β_i respectively be the smaller angles between the ith segment and the ith link, and between ith link and $(i+1)$th segment, for $i = 1,\ldots,k-1$. With this notation, the code length $C(P)$ of P can be decomposed into

$$\begin{aligned} C(P) = {} & C(z) + C(m) + C(S_1) + C(\alpha_1) \\ & + C(\text{direction of } \alpha_1) + C(L_1) + C(\beta_1) \\ & + C(\text{direction of } \beta_1) + C(S_2) + C(\alpha_2) \\ & + C(\text{direction of } \alpha_2) + C(L_2) + C(\beta_2) \\ & + C(\text{direction of } \beta_2) + \cdots + C(S_k). \end{aligned} \quad (3)$$

In order to proceed further, we make the following assumptions. Note that some of these assumptions were

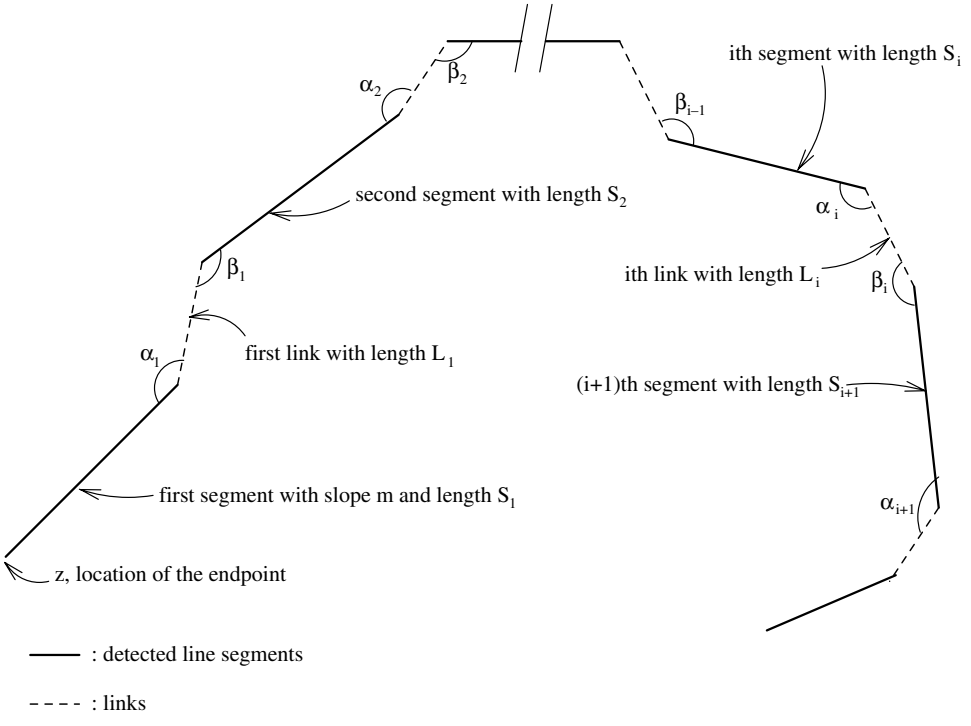

Fig. 1. Notation for polygonal line P.

also adopted by Lee and Talbot (1997), in which the MDL principle was applied to solve a simpler line segment joining problem.

(Al) The endpoint z of the first line segment is uniformly distributed in the space of the data, and the slope m of this same line segment is uniformly distributed in $[0, 2\pi)$. This is the same as stating that nothing is known about the first line segment, as its location and orientation are completely random.

(A2) The lengths $S_1, \ldots, S_k$ of the segments are independently and identically distributed (iid) as a log-normal distribution with unknown mean μ_S and unknown variance σ_S^2. That is, their common probability density function (pdf) is

$$f_S(s) = \frac{1}{s\sqrt{2\pi\sigma_S^2}} \exp\left\{-\frac{1}{2\sigma_S^2}(\ln s - \mu_S)^2\right\}, \quad s > 0.$$

The log-normal distribution is one of the most common statistical models for modeling the length of objects. It is because its domain is non-zero and it also possesses many desirable statistical properties (e.g., closed-form maximum likelihood estimators exist for its parameters).

(A3) The lengths $L_1, \ldots, L_{k-1}$ of the links are iid exponentials with unknown mean μ. That is, their common pdf is

$$f_L(l) = \frac{1}{\mu} \exp\left(\frac{-l}{\mu}\right), \quad l > 0.$$

Modeling L_i's with exponentials means that short links are encouraged.

(A4) The quantities $\pi - \alpha_i$ and $\pi - \beta_i$, $i = 1, \ldots, k-1$, are iid truncated exponentials with unknown mean v and truncation at π (e.g., see Johnson et al., 1994, Chapter 19). For convenience, we write $\gamma_i = (\pi - \alpha_i)$ and $\gamma_{k+i-1} = (\pi - \beta_i)$, $i = 1, \ldots, k-1$. It follows that the γ_i's are iid with the following common pdf:

$$f_\gamma(\theta) = \left\{1 - \exp\left(\frac{-\pi}{v}\right)\right\}^{-1} \frac{1}{v} \exp\left(\frac{-\theta}{v}\right), \quad 0 < \theta < \pi. \tag{4}$$

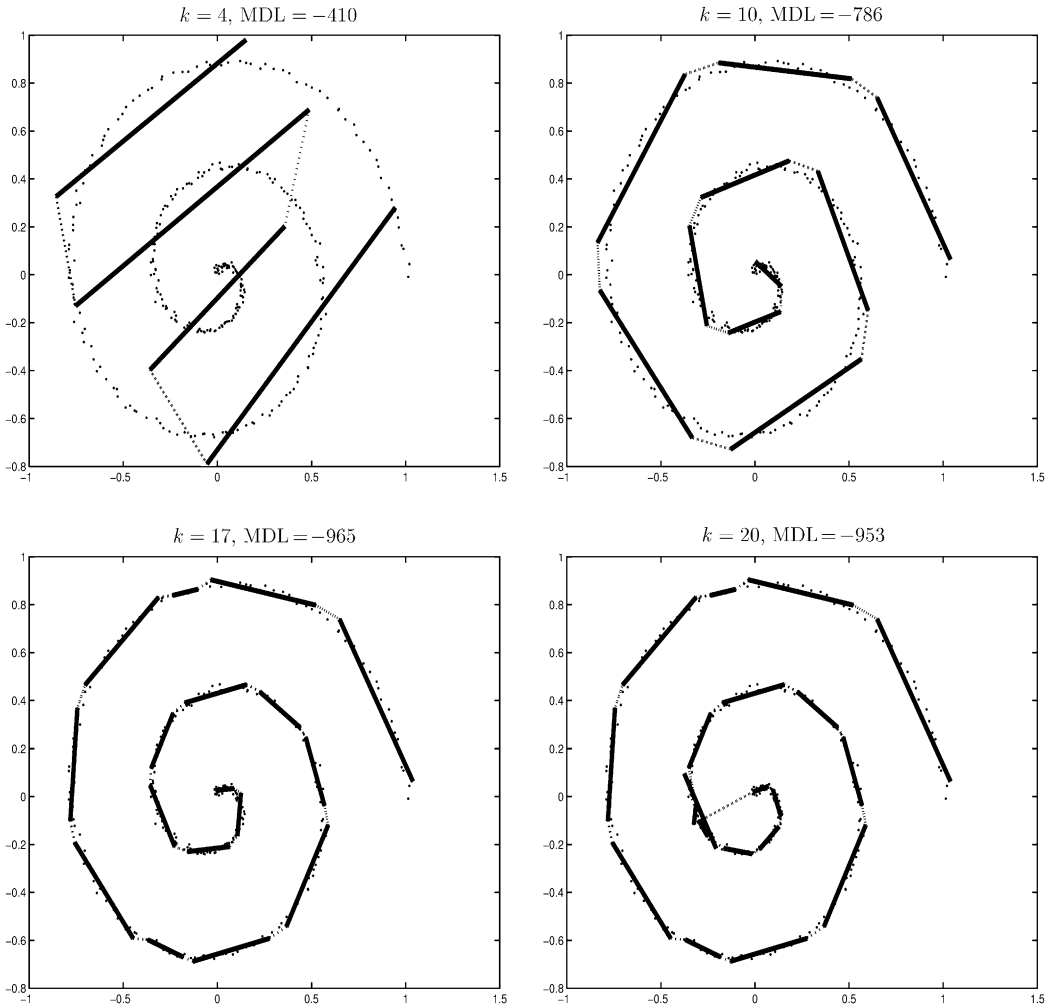

Fig. 2. Spiral: fitted line segments and links obtained by the k-segments algorithm for $k = 4, 10, 17, 20$.

Thus we prefer the angles α_i's and β_i's to be close to π, which in turns suggesting that we prefer the line segments and the links are well aligned.

Now if $\overline{L} = \sum L_i/(k-1)$ and $\bar{\gamma} = \sum \gamma_i/(2k-2)$, then it is shown in the Appendix that $C(L)$ can be well approximated by (upto a constant):

$$C(P) = 3(k-1) + (k-1)\log\overline{L} + \log(k-1) + \log k + \frac{k}{2}\log(2\pi+1) + \sum_{i=1}^{k}\log S_i + \frac{k}{2}\log\hat{\sigma}_S^2 + 2(k-1)\left[\log\left\{1-\exp\left(\frac{-\pi}{\hat{\nu}}\right)\right\} + \log\hat{\nu} + \frac{\bar{\gamma}}{\hat{\nu}}\right], \quad (5)$$

where

$$\hat{\nu} = \bar{\gamma} + \pi\left\{\exp\left(\frac{\pi}{\bar{\gamma}}\right) - 1\right\}^{-1}, \quad \hat{\mu}_S = \frac{1}{k}\sum_{i=1}^{k}\log S_i \quad \text{and}$$

$$\hat{\sigma}_S^2 = \frac{1}{k}\sum_{i=1}^{k}(\log S_i - \hat{\mu}_S)^2.$$

3.2. Code length calculation for $C(\{t_i\}_{i=1}^n)$ and $C(\{\|x_i - \hat{x}_i\|\}_{i=1}^n)$

Under our probabilistic model assumptions, the projections $\hat{x}_i$'s are independently and uniformly distributed along the principal curve. If the length of the polygonal line P is T, this uniform distribution assumption is equivalent to assuming that the arc length t_i's are iid uniform in $[0, T]$. Using (A.1) in the appendix, and ignoring negligible terms, we obtain

$$C(\{t_i\}_{i=1}^n) = n\log T.$$

Now for the code length of the difference $\|x_i - \hat{x}_i\|$'s. Under our model assumptions, these differences $\|x_i - \hat{x}_i\|$'s are iid $N(0,\sigma^2)$, in which the maximum likelihood estimate (MLE) for σ^2 is $\hat{\sigma}^2 = \frac{1}{n}\sum_{i=1}^n \|x_i - \hat{x}_i\|^2$. Now by applying (A.1) and dropping negligible terms, we have

$$C(\{\|x_i - \hat{x}_i\|\}_{i=1}^n) = \frac{1}{2}\log n + \frac{n}{2}\log\hat{\sigma}^2.$$

Notice that we have also ignored the code length that describes, for all i, whether x_i lies above or below the poly-

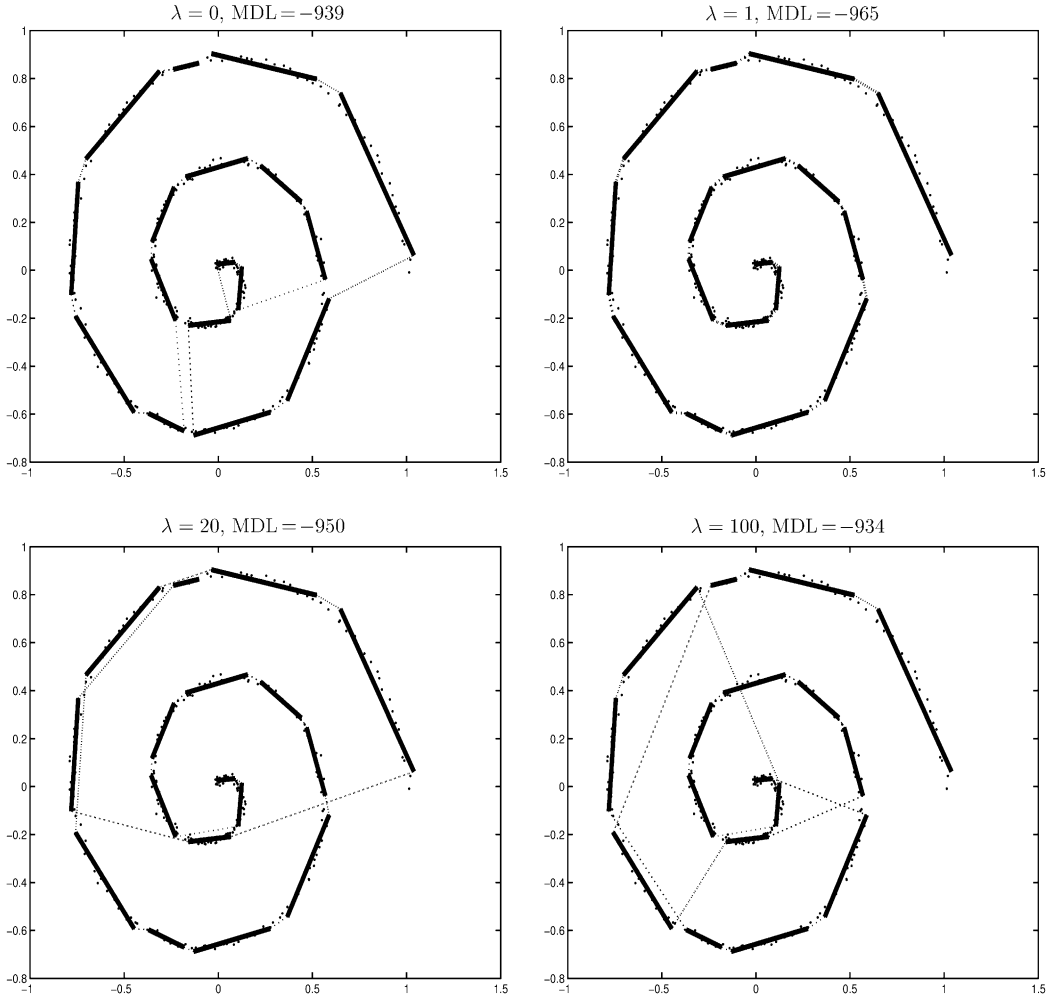

Fig. 3. Spiral: fitted line segments and links obtained by the k-segments algorithm for $\lambda = 0, 1, 20, 100$, all with $k = 17$.

gonal line. It is because for each x_i it takes $\log_2 2 = 1$ bit to encode this information and hence the total code length for all x_i's is n, which is a constant with respect to our minimization problem. Now, from (2), the overall code length for $C(\boldsymbol{x})$ is

$$\begin{aligned} C(\boldsymbol{x}) &= C(P) + C(\{t_i\}_{i=1}^n) + C(\{\|x_i - \hat{x}_i\|\}_{i=1}^n) \\ &= 3(k-1) + (k-1)\log\overline{L} + \log(k-1) + \log k \\ &\quad + \frac{k}{2}\log(2\pi+1) + \sum_{i=1}^{k}\log S_i + \frac{k}{2}\log\hat{\sigma}_S^2 \\ &\quad + 2(k-1)\left[\log\left\{1 - \exp\left(\frac{-\pi}{\hat{v}}\right)\right\} + \log\hat{v} + \frac{\bar{\gamma}}{\hat{v}}\right] \\ &\quad + n\log T + \frac{1}{2}\log n + \frac{n}{2}\log\hat{\sigma}^2. \end{aligned} \tag{6}$$

We propose to choose (k, λ) as the pair that minimizes (6). Notice that λ enters this expression through the two "angle-related" quantities $\bar{\gamma}$ and $\hat{v}$, as λ controls the angles between the line segments and the links. A direct comparison shows that this overall code length formula, (6), can be seen as a refinement of the likelihood function (8) of Verbeek et al. (2002).

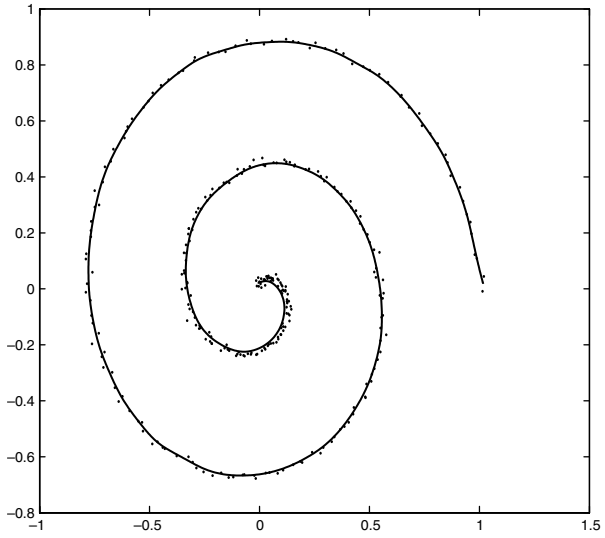

Fig. 4. Spiral: smoothed polygonal line for $k = 17$.

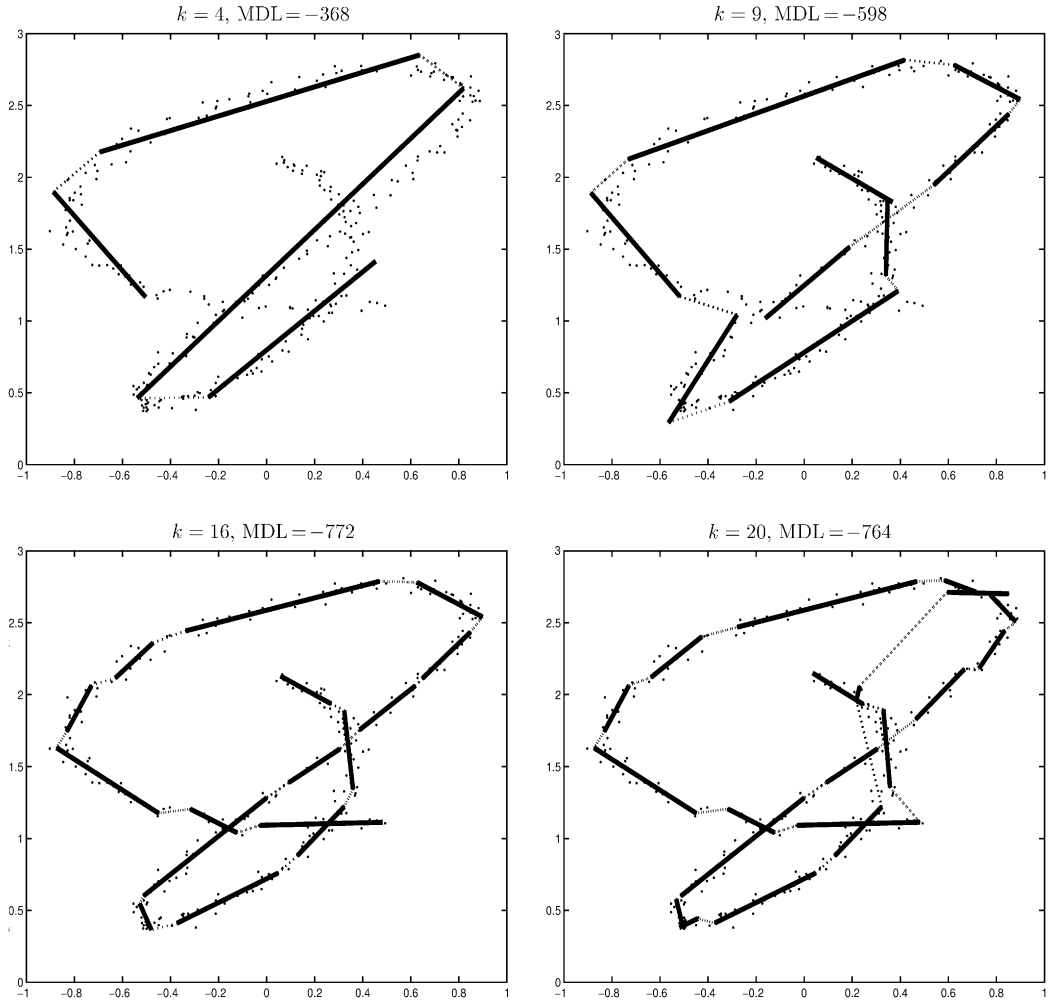

Fig. 5. cros3: fitted line segments and links obtained by the k-segments algorithm for $k = 4, 9, 16, 20$.

3.3. Practical minimization of (6)

Due to the co mplicated nature of (6), analytic minimization of this code length formula seems to be intractable. Therefore in practice we minimizes this code length formula (6) by conducting a 2D grid search in the space of (k, λ). That is, we first compute (6) for a set of different combinations of (k, λ) and then choose the pair that gives the smallest value of (6) as the final minimizer. In all our numerical experiments we computed 20×10 different combinations of (k, λ), where k ranged from 1 to 20 with a unit increment, while the 10 values of λ ranged from 0 to 100 with approximate equal spacing in the log scale. For all the examples to be reported in the next section, with an Intel Pentium M 1.6 GHz machine, it took 1000–1200 s to finish one minimization.

4. Numerical examples

In this section we assess the practical performance of the proposed method with two artificial examples. These two examples have been used by previous authors (e.g., Kegl et al., 2000 and Verbeek et al., 2002).

4.1. Spiral

The true curvilinear feature in this first example is a spiral. From this feature 300 spatial points x_i's were generated with noise standard deviation $\sigma = 0.01$. The k-segments algorithm was applied to compute the corresponding principal curves for different combinations of (k, λ). For illustrative purposes, the resulting principal curves for $k = 4, 10, 17, 20$, with their best λ values, are given in Fig. 2. In each subplot, the data points x_i's are shown as dots, the fitted line segments are denoted by thick lines, and the links are shown as dotted lines.

The MDL scores for the four cases when $k = 4, 10, 17, 20$ are -410, -786, -965 and -953, respectively. Thus MDL suggests that $k = 17$ is the best choice. One can see that, for $k = 4$, the number of line segments is not large enough to capture the shape characteristics of the spiral. For $k = 10, 17, 20$, it seems that the overall shape is well captured by the linked polygonal lines. However, one could

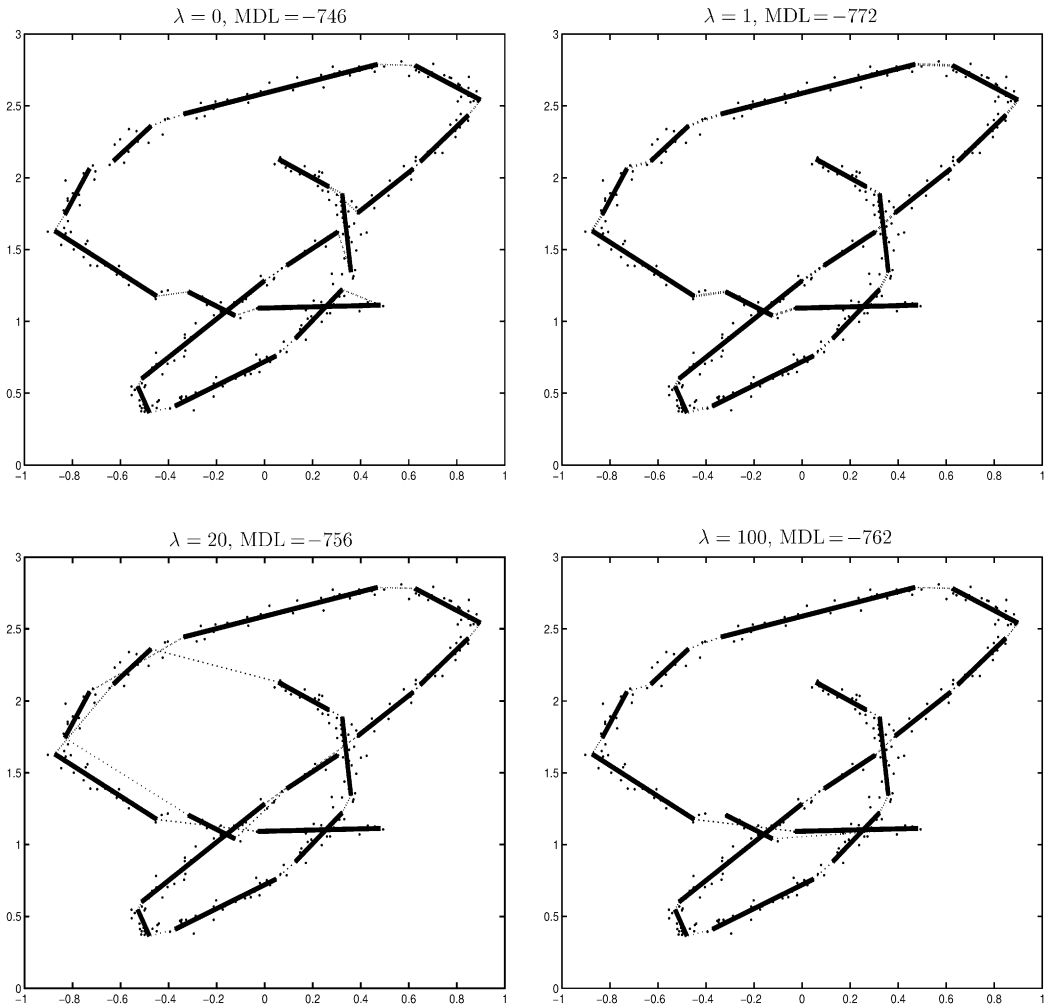

Fig. 6. cros3: fitted line segments and links obtained by the k-segments algorithm for $\lambda = 0, 1, 20, 100$, all with $k = 16$.

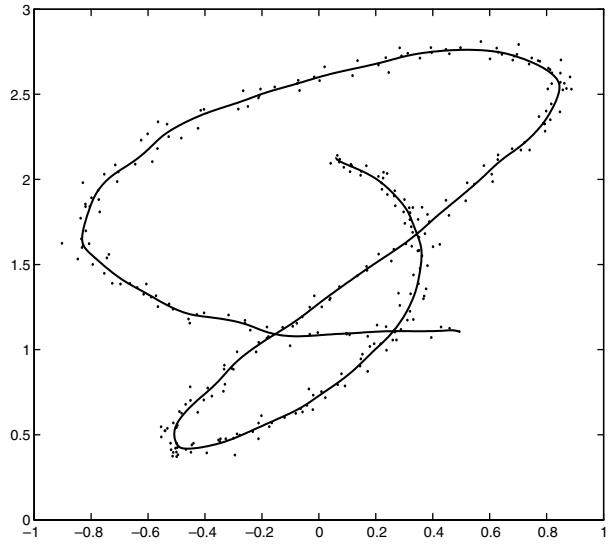

Fig. 7. cros3: smoothed polygonal line for $k = 16$.

see that $k = 10$ underfits while $k = 20$ overfits the data. To demonstrate the effect of λ on the linked polygonal lines, displayed in Fig. 3 are four sets of linked polygonal lines, all obtained with $k = 17$ but with different values of λ. One can see that the MDL choice ($\lambda = 1$) gives the best visual result. Lastly, the linked polygonal line for the best combination of (k, λ) is further smoothed to obtain a smooth representation of the spiral. This smoothed polygonal line is shown in Fig. 4.

4.2. Self-intersecting

In this second example the true curvilinear feature is self-intersecting, and is termed cros3 by Verbeek et al. (2002). Its coordinates are given by

$$[\sqrt{t}\{0.1 + \sin(4\pi t + 0.4)\}, t + 1.1 + \cos(3\pi t + 0.1)], \quad t \in [0, 1].$$

Three hundred data points were generated with $\sigma = 0.03$. As before, the k-segments algorithm was applied to compute different principal curves for different combinations of (k, λ). Again, for illustrative purposes, the resulting principal curves for $k = 4, 9, 16, 20$ are displayed in Fig. 5. The corresponding MDL scores are, respectively, -368, -598, -772 and -764. For $k = 4$ and $k = 9$, the resulting linked polygonal lines outline some major characteristics of the curve, but are still insufficient to represent the whole. For $k = 20$, the linked polygonal line does capture the overall shape of the curve, but it also includes some spurious structures, such as the extra line segment in the upper right corner. Thus it appears that $k = 16$ is the best reconstruction, which is the one suggested by MDL. As in the previous example, linked polygonal lines correspond to different λ's are also obtained; see Fig. 6. Finally, Fig. 7 depicts the smoothed polygonal line obtained from the best combination of (k, λ).

5. Conclusion

In this paper we studied the k-segments algorithm proposed by Verbeek et al. (2002) for computing principal curves. In particular we developed an automatic method for choosing the "free" tuning parameters (k, λ) of this k-segments algorithm. The idea was to first pose this choosing problem as a statistical model selection problem, and then apply the minimum description length principle to derive a solution. Numerical experiments were also performed for demonstrating the effectiveness of this new parameter selection method. Possible extensions of the present work include generalizing the current selection method to the cases when there are more than one principle curve in the space, and/or when background noise are present.

Acknowledgements

The authors are most grateful to the referees for their constructive comments. The work of Lee was partially supported by the National Science Foundation under Grant No. 0203901.

Appendix A. Derivation of code length expression (5)

This appendix derives the code length expression (5). First we present the following useful result from Rissanen (1989) (see also Lee, 2001). Suppose one would like to encode N iid observations $\boldsymbol{y} = (y_1, \ldots, y_N)$ generated from the pdf $f(y; \theta_1, \ldots, \theta_d)$ with d unknown parameters $\theta_1, \ldots, \theta_d$. If these θ_i's are known, then classical coding results of Shannon suggest that $C(\boldsymbol{y}) = -\sum_{i=1}^{N} \log_2 f(y_i; \theta_1, \ldots, \theta_d)$. Now if the θ_i's are unknown, one could first estimate them from the data $\boldsymbol{y}$, and then apply Shannon's results with the unknown θ_i's replaced by their estimates $\hat{\theta}_i$'s. Under this situation, Rissanen (1989, pp. 55–56) demonstrates that an maximum likelihood estimate (MLE) computed from Ndata points can be effectively encoded by $\frac{1}{2}\log_2 N$ bits. Thus the code length for each $\hat{\theta}_i$; is $\frac{1}{2}\log_2 N$ bits, and the overall code length for $\boldsymbol{y}$ is

$$C(\boldsymbol{y}) = \frac{d}{2}\log_2 N - \sum_{i=1}^{N} \log_2 f(y_i; \hat{\theta}_1, \ldots, \hat{\theta}_d). \tag{A.1}$$

Code lengths for z and m: Since z is assumed to be uniformly distributed in the space of $\boldsymbol{x}$, and m is assumed to be uniform from $[0, 2\pi)$, their code lengths are the same for different values of (k, λ). Therefore their values will have no effect on the minimization and hence are omitted from Expression (5).

Code length for S_i's: from Assumption A2 one can see that (A.1) can be directly applied to calculate the code length for S_i's. There are two unknown parameters, μ_S and σ_S^2, and their MLEs are respectively

$$\hat{\mu}_S = \frac{1}{k}\sum_{i=1}^{k} \ln S_i \quad \text{and} \quad \hat{\sigma}_S^2 = \frac{1}{k}\sum_{i=1}^{k} (\ln S_i - \hat{\mu}_S)^2.$$

Since both of these MLEs are calculated from k data points, their code length is $2 \times \frac{1}{2}\log_2 k = \log_2 k$. This corresponds to the first term of (A.1). For the second term of (A.1), one calculates

$$\begin{aligned}
&-\sum_{i=1}^{k} \log_2 f_S(S_i; \hat{\mu}_S, \hat{\sigma}_S^2) \\
&= -\sum_{i=1}^{k} \log_2 \left[\frac{1}{S_i \sqrt{2\pi\hat{\sigma}_S^2}} \exp\left\{ -\frac{1}{2\hat{\sigma}_S^2} (\ln S_i - \hat{\mu}_S)^2 \right\} \right] \\
&= \sum_{i=1}^{k} \log_2 S_i + \frac{k}{2}\log_2(2\pi\hat{\sigma}_S^2) + \frac{1}{2\hat{\sigma}_S^2} \sum_{i=1}^{k} (\ln S_i - \hat{\mu}_S)^2 \\
&= \sum_{i=1}^{k} \log_2 S_i + \frac{k}{2}\log_2(2\pi\hat{\sigma}_S^2) + \frac{1}{2\hat{\sigma}_S^2} k\hat{\sigma}_S^2 \\
&= \sum_{i=1}^{k} \log_2 S_i + \frac{k}{2}\log_2(2\pi) + \frac{k}{2}\log_2(\hat{\sigma}_S^2) + \frac{k}{2}.
\end{aligned}$$

Combining these two terms one obtains

$$\sum_{i=1}^{k} C(S_i) = \log_2 k + \frac{k}{2}(\log_2 2\pi + 1) + \frac{k}{2}\log_2 \hat{\sigma}_S^2 + \sum_{i=1}^{k} \log_2 S_i.$$

Code length for L_i's: as similar to those S_i's. Assumption A3 enables us to apply (A.1) to calculate the code length for the L_i's. The only unknown parameter in the common pdf (exponential) of the L_i's is μ, which can be estimated by $\hat{\mu} = \overline{L} = \sum L_i/(k-1)$. Notice that $\hat{\mu}$ is computed from $k-1$ data points, so it needs $\frac{1}{2}\log_2(k-1)$ bits to encode. With (A.1), a straightforward calculation gives

$$\sum_{i=1}^{k-1} C(L_i) = \frac{1}{2}(k-1) + (k-1)(\log_2 \overline{L} + 1).$$

Code length for α_i's, β's and their directions: we begin with computing the code length for the α_i's and the β_i's. First recall the following definition stated in Assumption A4: $\gamma_i = \pi - \alpha_i$ and $\gamma_{k+i-1} = \pi - \beta_i$, $i = 1, \ldots, k-1$. Thus a complete knowledge of $\{\alpha_i, \beta_i\}_{i=1}^{k-1}$ implies a complete knowledge of $\{\gamma_i\}_{i=1}^{2k-2}$ and vice versa. Hence it is enough to only encode the γ_i's. From Assumption A4 one can see that the γ_i's are iid with common pdf (4), and (A.1) can be applied to calculate the required code length. There is only one unknown parameter v in pdf (4), whose MLE $\hat{v}$ is given by the solution of the equation (e.g., see Johnson et al., 1994, Chapter 19):

$$\hat{v} = \bar{\theta} + \pi\left\{ \exp\left(\frac{\pi}{\hat{v}}\right) - 1 \right\}^{-1}.$$

However, for the ease of computation and by using the argument $\hat{v} \approx 0 \Rightarrow \exp(\frac{\pi}{v})$ is large $\Rightarrow \hat{v} \approx \bar{\theta}$, we approximate the MLE of v by

$$\hat{v} = \bar{\theta} + \pi\left\{ \exp\left(\frac{\pi}{\bar{\theta}}\right) - 1 \right\}^{-1}.$$

Notice that $\hat{v}$ is estimated from $2k-2$ data points, hence it requires $\frac{1}{2}\log_2(2k-2)$ bits to encode. A straightforward application of (A.1) gives

$$\begin{aligned}
\sum_{i=1}^{k-1} \{C(\alpha_i) + C(\beta_i)\} = \sum_{i=1}^{2k-2} C(\gamma_i) &= \frac{1}{2}\log_2(2k-2) + (2k-2) \\
&\times \left[\log_2\left\{ 1 - \exp\left(\frac{-\pi}{\hat{v}}\right) \right\} + \log_2 \hat{v} + \frac{\bar{\theta}}{\hat{v}} \right].
\end{aligned}$$

The last part of the code length that we need is the part for the directions of α_i's and β_i's, Each direction, either upward or downward, requires 1 bit to encode and thus altogether it requires $2k-2$ bits to encode all the directions. Therefore we have

$$\begin{aligned}
&\sum_{i=1}^{k-1} \{C(\alpha_i) + C(\beta_i) + C(\text{direction of } \alpha_i) + C(\text{direction of } \beta_i)\} \\
&= \frac{1}{2}\log_2(2k-2) + (2k-2) \\
&\quad \times \left[\log_2\left\{ 1 - \exp\left(\frac{-\pi}{\hat{v}}\right) \right\} + \log_2 \hat{v} + \frac{\bar{\theta}}{\hat{v}} \right] + (2k-2).
\end{aligned}$$

Now if we combine the above relevant code length expressions, ignore constant terms and change $\log_2$ to log (natural log), we obtain expression (5) for $C(P)$.

References

Bishop, C.M., Svensen, M., Williams, C.K.I., 1998. GTM: The generative topographic mapping. Neural Comput. 10, 215–234.

Cheng, Z.G., Chen, M., Liu, Y.C., 2004. A robust algorithm for image principal curve detection. Pattern Recognition Lett. 25, 1303–1313.

Delicado, P., Huerta, M., 2003. Principal curves of oriented points: Theoretical and computational improvements. Comput. Statist. 18, 293–315.

Fritzke, B., 1994. Growing cell structure—a self-organizing network for unsupervised and supervised learning. Neural Networks 7, 1441–1460.

Hansen, M., Yu, B., 2000. Wavelet thresholding via MDL for natural images. IEEE Trans. Inform. Theory 46, 1778–1788.

Hastie, T., Stuetzle, W., 1989. Principal curves. J. Amer. Statist. Assoc. 84, 502–516.

Johnson, N.L., Kotz, S., Balakrishnan, N., 1994. Continuous Univariate Distributions, second ed., vol. 1. John Wiley & Sons, Inc.

Kegl, B., Krzyzak, A., Linder, T., Zeger, K., 2000. Learning and design of principal curves. IEEE Trans. Pattern Anal. Machine Intell. 22, 281–297.

Kohonen, T., 1995. Self-Organizing Maps. Springer, Berlin.

Lee, T.C.M., 2001. An introduction to coding theory and the two-part minimum description length principle. Internat. Statist. Rev. 69, 169–183.

Lee, T.C.M., Talbot, H., 1997. Automatic reconnection of linear segments by the minimum description length principle. In: Proc. of DICTA 97, Digital Image Computing: Techniques and Applications.

Rissanen, J., 1989. Stochastic Complexity in Statistical Inquiry. World Scientific, Singapore.

Tibshirani, R., 1992. Principal curves revisited. Statist. Comput. 2, 183–190.

Verbeek, J.J., Vlassis, N., Krose, B., 2002. A k-segments algorithm for finding principal curves. Pattern Recognition Lett. 23, 1009–1017.

Xie, J., Zhang, D., Xu, W., 2004. Spatially adaptive wavelet denoising using the minimum description length principle. IEEE Trans. Image Process. 13, 179–187.