Available online at www.sciencedirect.com
SCIENCE DIRECT®
Pattern Recognition 38 (2005) 1079 – 1085
PATTERN RECOGNITION
THE JOURNAL OF THE PATTERN RECOGNITION SOCIETY
www.elsevier.com/locate/patcog

A bottom-up algorithm for finding principal curves with applications to image skeletonization

Xiabi Liu, Yunde Jia*

Department of Computer Science and Engineering, Beijing Institute of Technology, Beijing 100081, PR China

Received 29 April 2004; received in revised form 29 November 2004; accepted 29 November 2004

Abstract

This paper proposes a new method for finding principal curves from data sets. Motivated by solving the problem of highly curved and self-intersecting curves, we present a bottom-up strategy to construct a graph called a principal graph for representing a principal curve. The method initializes a set of vertices based on principal oriented points introduced by Delicado, and then constructs the principal graph from these vertices through a two-layer iteration process. In inner iteration, the kernel smoother is used to smooth the positions of the vertices. In outer iteration, the principal graph is spanned by minimum spanning tree and is modified by detecting closed regions and intersectional regions, and then, new vertices are inserted into some edges in the principal graph. We tested the algorithm on simulated data sets and applied it to image skeletonization. Experimental results show the effectiveness of the proposed algorithm.

Keywords: Principal curves; Principal oriented points; Image skeletonization; Kernel smoother; Minimum spanning tree

1. Introduction

Principal component analysis is widely used in dimension reduction, and feature extraction. Principal curves are nonlinear generalizations of principal components and have received significant attention since their introduction by Hastie and Stuetzle [1]. Considerable work has been reported about applications of principal curves, such as, shape detection and object identification [2,3], gradient analysis in ecology [4], image skeletonization [5–7], feature extraction and pattern classification [8,9], speech recognition [10], and forecasting [11].

There have been several different definitions of principal curves. The earliest one by Hastie and Stuetzle [1] emphasizes the self-consistency property of principal curves, which means that each point of a principal curve is the average of all data projecting there. Tibshirani [12] gave his definition in terms of the mixture probability model in which a distribution is decomposed into a latent variable distribution on a curve and a conditional distribution given the latent variable value. Kégl et al. [6] defined a principal curve as a curve which minimizes the expected squared distance from data to their projection on the curve over a class of curves with bounded length. Sandilya and Kulkarni [13] provided a similar definition, but they constrained total turn instead of length. Delicado [14] introduced the notion of principal oriented points (POPs) and made principal curves visit only POPs.

Based on these definitions, a few methods for finding principal curves from data sets have been proposed. Hastie and Stuetzle [1] proposed an alternation between projecting data onto the curve and estimating conditional expectations on projectors by the scatter smoother or the spline smoother. Banfield and Raftery [2] modified the Hastie–Stuetzle

* Corresponding author. Tel.: +86 10 689 40955; fax: +86 10 689 45011.

E-mail addresses: liuxiabi@bit.edu.cn (X. Liu), yjiar@bit.edu.cn (Y. Jia).

0031-3203/$30.00
doi:10.1016/j.patcog.2004.11.016

method and used the projection residual of the data, instead of the data themselves, to estimate conditional expectations for reducing both bias and variance. Tibshirani [12] used the EM algorithm to maximize the log-likelihood of the observation implied by his mixture model under the Gaussian assumption. Verbeek et al. [15] proposed a k-segments algorithm which incrementally combines local line segments into the polygonal line to achieve an objective similar to Tibshirani's. Kégl et al. [6] presented the polygonal line algorithm which starts with an initial polygonal line, adds a new vertex to the polygonal line at each iteration, and updates the positions of all vertices so that the value of a penalized distance function is minimized. Singh et al. [5] used the batch formulation of the self-organizing mapping (SOM) to obtain principal curves. Delicado [14] found the principal oriented points one by one and orderly linked them to estimate principal curves. Other related techniques, such as the generative topographic mapping and the growing cell structures, can also be used to find approximations to principal curves [15].

For highly curved or self-intersecting curves such as spiral-shaped curves [6,15], existing methods did not work well. Verbeek et al. [15] attempted to solve this problem by combining line segments which were optimized to minimize the total squared distance of all points to their closest segments into a polygonal line. The first principal component of all data is often used as the initial estimation of the principal curve when lacking the prior knowledge. Unfortunately it is a bad initialization for a highly curved or self-intersecting curve. So it is necessary to consider the local feature of a principal curve from the beginning. In this paper, we present a bottom-up strategy to construct a graph (called a principal graph similar to that at Kégl et al. [6]) for representing a principal curve. Instead of starting with a simple topology such as the first principal component and then increasing its complexity iteratively, we directly span the sufficient complex topology and then refine it iteratively. In our algorithm, POPs in local areas are founded as initial candidate points on a principal curve; we then take these points as a set of vertices, and perform a two-layer iteration process from it to construct a principal graph.

This paper is organized as follows. Section 2 introduces the definitions of principal curves and principal oriented points. Section 3 describes the bottom-up algorithm in detail. Section 4 discusses the test results on simulated data sets and applications to image skeletonization. We conclude the paper with a discussion in Section 5.

2. Principal curves and principal oriented points

A principal curve always has the self-consistency property [1]. Let X denote a random vector in R^d, and $f(\lambda)$ denote a smooth (infinitely differentiable) curve in R^d parameterized by $\lambda \in R^1$. The projection index $\lambda_f : R^d \to R^1$ is defined as

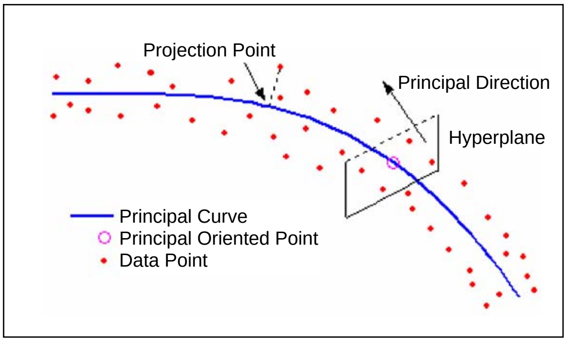

Fig. 1. Illustration of principal curves and principal oriented points.

$$\lambda_f(X) = \sup_\lambda \left\{ \lambda : \|X - f(\lambda)\| = \inf_\mu \|X - f(\mu)\| \right\}. \quad (1)$$

The curve $f(\lambda)$ is self-consistent if

$$f(\lambda) = E(X|\lambda_f(X) = \lambda) \quad (2)$$

for almost all λ. Intuitively, the self-consistency means that a principal curve passes through the "middle" of a distribution and each point of it is the average (under the distribution) of all points that project there, as illustrated in Fig. 1.

According to the self-consistency mentioned above, we cannot know whether a point is self-consistent unless we know the whole curve to which this point belongs. Delicado [14] discussed the self-consistency of a single point and established the definition of principal oriented points (POPs) based on the property of the first principal component for normal distribution that can be stated as the projection of the normal random variable onto the hyperplane orthogonal to the first principal component has the lowest total variance among all the projected variable onto any hyperplane.

Let $b \in S^{d-1} = \{w \in R^d | \|w\| = 1\}$, $H(X, b)$ be hyperplane orthogonal to b passing through X : $H(X, b) = \{Y \in R^d | (Y - X)^t b = 0\}$, $u(X, b)$ and $\phi(X, b)$ be the conditional expectation and the total variance of random variables on $H(X, b)$, respectively: $u(X, b) = E(Y|Y \in H(X, b))$, and $\phi(X, b) = TV(Y|Y \in H(X, b))$. The b for achieving the infimum of $\phi(X, b)$ is defined as the principal direction of X and denoted by $b^*(X)$: $b^*(X) = \arg\min_{b \in S^{d-1}} \phi(X, b)$. The corresponding $u(X, b)$ is

$$u^*(X) = u(X, b^*(X)). \quad (3)$$

Fixed points of $u^*(X)$ are defined as principal oriented points which are denoted by $\Gamma(X)$: $\Gamma(X) = \{Y \in R^d | Y \in u^*(X)\}$, as shown in Fig. 1.

Delicado [14,16] proposed an algorithm to find a corresponding POP starting with an arbitrary point in the data set. Let $\tilde{u}^*(X)$ be the sample version of the function $u^*(X)$ in the formulation (3). X_0 be an arbitrary point in the data set.