Parametric Representations for Nonlinear Modeling of Visual Data

Ying Zhu† Dorin Comaniciu‡ Visvanathan Ramesh‡ Stuart Schwartz†

†Electrical Engineering
Princeton University
Princeton, NJ 08544

‡Imaging & Visualization Department
Siemens Corporate Research
755 College Road East, Princeton, NJ 08540

Abstract

Accurate characterization of data distribution is of significant importance for vision problems. In many situations, multivariate visual data often spread into a nonlinear manifold in the high-dimensional space, which makes traditional linear modeling techniques ineffective. This paper proposes a generic nonlinear modeling scheme based on parametric data representations. We build a compact representation for the visual data using a set of parameterized basis (wavelet) functions, where the parameters are randomized to characterize the nonlinear structure of the data distribution. Meanwhile, a new progressive density approximation scheme is proposed to obtain an accurate estimate of the probability density, which imposes discrimination power on the model. Both synthetic and real image data are used to demonstrate the strength of our modeling scheme.

1. Introduction

We address the problem of learning parametric models from multivariate data. In many pattern recognition and vision applications, an interesting pattern is measured or visualized through multivariate data such as time signals and images. Its random occurrences are described as scattered data points in the high-dimensional space. For accurate representation and effective use of the decisive information, it is important to explore the intrinsic low dimensionality of the scattered data and to obtain an accurate statistical model for the data distribution. The general procedure of parametric modeling approximates the data distribution with a family of parameterized density models and then estimates model parameters for the best fit to the data.

Among the commonly used techniques, principal components analysis (PCA) [1,2] adopts a low-dimensional linear representation and approximates the multivariate data by a low rank Gaussian density model. Linear factor analysis [3] approximates class distribution as a Gaussian distribution with structured covariance matrix. These linear modeling schemes are capable of representing the data distributions with ellipsoidal shape, but they are unable to handle the situations where data samples spread into a nonlinear manifold that is no longer Gaussian. The nonlinear structure of multivariate data distribution is not unusual even when the internal decisive variables of the pattern have a unimodal distribution. For example, images of an object under varying poses form a nonlinear manifold in the high-dimensional space. The expression variation can lead to structural nonlinearity on the data distribution of single view face images. Similar situation happens when we model the images of cars with a variety of outside designs. One way to handle the nonlinearity is through multimodal approximation using a mixture density model [4, 5]. Alternative methods have been proposed to directly identify the nonlinear principal manifold [6, 7, 22, 23]. However, no probabilistic model is associated with these approaches.

This paper presents a new statistical modeling scheme that not only characterizes the nonlinear structure of the data distribution but also represents it with a probabilistic distribution model. The modeling scheme is built by defining a parametric function representation for multivariate data. We statistically model the random function parameters to obtain a density estimate of the data distribution. Compared to other nonlinear modeling schemes, the probabilistic distribution model obtained by our scheme provides a likelihood measure therefore has the discrimination power essential for pattern identification.

The paper is organized as follows. In section 2, we introduce the parametric function representation for multivariate data. In section 3 and 4, a new progressive density approximation scheme is proposed to obtain the probabilistic distribution function. Experimental results on both synthetic and real data are presented in section 5. We finish with conclusions and discussions in sections 6.

2. Parametric function representation

In addition to the traditional vector representation $\mathbf{y} \in R^n$ for *n*-dimensional multivariate data, we associate a smooth function for every data sample. A length-*n* vector $\mathbf{y} = [y_1, \cdots y_n]^T$ is associated with a smooth function $y(t) \in L^2 : R^1 \to R^1 (t \in R^1)$ interpolated from its discrete components $y(t_i) = y_i (i = 1, \cdots, n)$. When the multivariate data are images, functions defined on R^2, $y(\mathbf{t}) \in L^2 : R^2 \to R^1 (\mathbf{t} = (t_1, t_2) \in R^2)$, are adopted for the representation. The function association applies a smoothing process to the discrete data components and is unique when only smooth functions are involved. In practice, these functions may be immediately available from data generating process. As a result, in the function space L^2, there is a counterpart of the scattered multivariate data located in vector space.

Compared with linear representation in PCA or Gaussian mixtures model, the advantage of function association lies in its ease to handle the non-linearity by parametric effects embedded in the data. It has intensive use in functional data analysis [8,9,19] for recovering principal curves in 1D signals. In [6], spline functions are used to parameterize the nonlinear principal manifold of an arbitrary data distribution. In [10], the parametric representation also provides a high-level semantic description of the pattern.

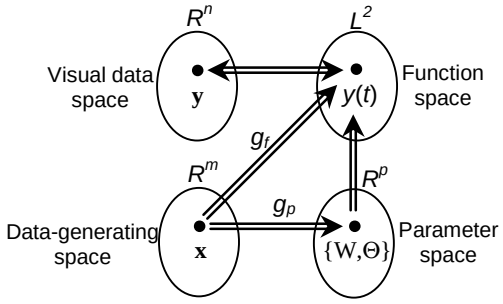

Figure 1. Parametric function representation through space mappings. Multivariate data **y** is represented in function form $y(t)$ parameterized by $\{W, \Theta\}$.

Figure 1 illustrates the idea of parametric function representation. Let $\{b_\theta\}$ be a base set for the function space L^2. The basis function $b_\theta(\mathbf{t}) = b(\mathbf{t};\theta): R^d \to R \quad (\mathbf{t} \in R^d)$ is parameterized and indexed by parameter θ, where θ can take a continuous range of values. Examples of such base sets include wavelets, splines and trigonometric functions. Any function $y(\mathbf{t}) \in L^2$ can be closely approximated by a sufficient number (N) of basis functions,

$$y(\mathbf{t}) \cong [w_1, \cdots w_N] \cdot \begin{bmatrix} b(\mathbf{t};\theta_1) \\ \vdots \\ b(\mathbf{t};\theta_N) \end{bmatrix} \tag{1}$$

The basis function $b(\mathbf{t};\theta)$ is generally nonlinear in θ. Once the basis b is chosen, the function $y(\mathbf{t})$ is completely determined by the linear parameter set $\{w_1, \cdots, w_N\}$ and the nonlinear set $\{\theta_1, \cdots, \theta_N\}$. The vector notations W_N, Θ_N, and $\mathbf{\Theta}_N$ for the linear, nonlinear and overall parameter sets, are used in the following discussions, where N denotes the number of basis functions involved.

$$W_N = \begin{bmatrix} w_1 \\ \vdots \\ w_N \end{bmatrix}; \quad \Theta_N = \begin{bmatrix} \theta_1 \\ \vdots \\ \theta_N \end{bmatrix}; \quad \mathbf{\Theta}_N = \begin{bmatrix} \mathbf{\Theta}_{N-1} \\ w_N \\ \theta_N \end{bmatrix} \tag{2}$$

To study data distribution, we assume $\mathbf{x} \in R^m$, in a vector form, to be the underlying quantities with minimum dimension (intrinsic dimensionality) that govern the data-generating process. Each observed data point reflects an occurrence of $\mathbf{x}$. Our goal is to statistically characterize $\mathbf{x}$ using the observed data $\mathbf{y}$. With parametric function association, the data-generating process can be formulated as a mapping g_f from the domain of $\mathbf{x}$ to the function space L^2, $g_f : R^m \to L^2$, or equivalently, as a mapping g_p from the domain of $\mathbf{x}$ to the domain of parameter set denoted by R^p, $g_p : R^m \to R^p$,

$$g_p(\mathbf{x}) = [W_N(\mathbf{x}), \Theta_N(\mathbf{x})]^T \tag{3}$$

By defining the matrix

$$\mathbf{B}(\Theta_N) = \left[\mathbf{b}(\theta_1), \cdots, \mathbf{b}(\theta_N)\right] \quad \text{where} \quad \mathbf{b}(\theta_i) = \begin{bmatrix} b(\mathbf{t}_1;\theta_i) \\ \vdots \\ b(\mathbf{t}_n;\theta_i) \end{bmatrix} \tag{4}$$

the multivariate data $\mathbf{y}$ is related to $\mathbf{x}$ as

$$\begin{aligned} \mathbf{y} &= [y_1, \cdots, y_n]^T \\ &= \left[\mathbf{b}(\theta_1(\mathbf{x})), \cdots, \mathbf{b}(\theta_N(\mathbf{x}))\right] \cdot W_N(\mathbf{x}) + \mathbf{n} \\ &= \mathbf{B}(\Theta_N(\mathbf{x})) \cdot W_N(\mathbf{x}) + \mathbf{n} \end{aligned} \tag{5}$$

where $\mathbf{n}$ is introduced to account for the noise in the observed data as well as the representation error. By choosing a proper type of basis functions, (5) defines a compact representation for multivariate data, and modeling the data distribution can be resolved through modeling the parameter set $\mathbf{\Theta}_N$.

3. Statistical modeling through progressive distribution approximation

Based on the parametric function representation (5), here we discuss algorithms and criteria for modeling the data distribution. In most situations with a single pattern involved, the governing factor $\mathbf{x}$ is likely unimodal although the observed data $\mathbf{y}$ may disperse into a nonlinear manifold due to parametric effects. Our discussion is primarily restricted to such internally unimodal data clusters. Figure 2 shows a toy example. The observed multivariate data $\mathbf{y} = [y_1, \cdots, y_n]^T$ consists of n equally spaced samples from a random process $y(t; w, \theta)$,

$$\begin{aligned} &y_i = y(t_i; w, \theta); \qquad (t_i = i \cdot T) \\ &y(t; w, \theta) = w \cdot b(t; \theta); \qquad (\theta = (s, t_0)) \\ &b(t; \theta) = (t - t_0) exp(-\frac{t - t_0}{s}) \end{aligned} \tag{6}$$

where w, s and t_0 are random variables with normal distributions. The data-generating process is characterized by these three intrinsic parameters. Figure 2(a) plots a few data samples in the form of time signals. Figure 2(b) shows the nonlinear structure of the data distribution by plotting the data projections on their first two linear principal components. The linear PCA and multimodal approximation are either incapable of or inefficient in modeling such distribution, even though the distribution of the internal variables $\{w, s, t_0\}$ is simply Gaussian. Such phenomenon is familiar to many situations where the visual data is generated by a common pattern and bears similar features up to small

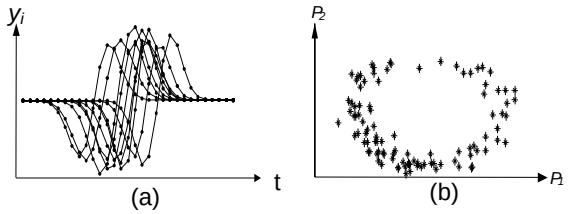

Figure 2. Nonlinearly distributed manifold. (a) Curve samples. (b) 2D visualization of the data distribution.

random deformation.

The framework of our modeling scheme is shown in Figure 3. The basic idea is to characterize the nonlinear structure of internally unimodal distributions by modeling the parametric effects. Equation (6) defines a single wavelet function denoted as the Derivative of Gaussian (DOG) function. In practice, visual data is more complicated and requires more basis functions to encode (1). These basis functions are parameterized with randomized parameters to accommodate random deformation. Wavelet functions are primarily used in this work for compact encoding of visual data. As the basis functions are gradually introduced, a sequence of density estimates, which approach the true data distribution, is also obtained.

3.1. Statistical modeling

From (3), both W_N and Θ_N are determined by $\mathbf{x}$ through the mapping g_p. Assuming that $\mathbf{x}$ is well modeled by normal distribution and the mapping g_p is smooth, the linearization of $W_N(\mathbf{x})$ and $\Theta_N(\mathbf{x})$ is valid around the mean of $\mathbf{x}$.

$$\begin{aligned} W_N(\mathbf{x}) &= W_{N,0} + A_{W,N} \cdot \mathbf{x} \\ \Theta_N(\mathbf{x}) &= \Theta_{N,0} + A_{\Theta,N} \cdot \mathbf{x} \end{aligned} \tag{7}$$

In practice, if prior knowledge shows that the assumption of normality or linearization does not hold well, we can always replace the normal distribution with a more suitable unimodal distribution form for $\mathbf{x}$ and include higher order terms in (7). The following discussion remains the same. However, to simplify the analysis, we keep normal distribution as nominal distribution for $\mathbf{x}$ and assume that linearization (7) is valid. Consequently, multivariate data $\mathbf{y}$ is effectively modeled by a jointly Gaussian distribution of W_N and Θ_N through (5).

$$p(\mathbf{y}) = \int_{W_N \Theta_N} p(W_N, \Theta_N) \cdot p(\mathbf{y} \mid W_N, \Theta_N) dW_N d\Theta_N \tag{8}$$

Based on this conclusion, the following discussion is devoted to finding the particular distribution from the family defined by (8) that best fits the observed data set.

Assume that $\mathbf{n}$ (5) is white Gaussian noise with zero mean and variance σ_N^2 and $\{W_N(\mathbf{x}), \Theta_N(\mathbf{x})\}$ has joint mean μ_N and covariance matrix Σ_N,

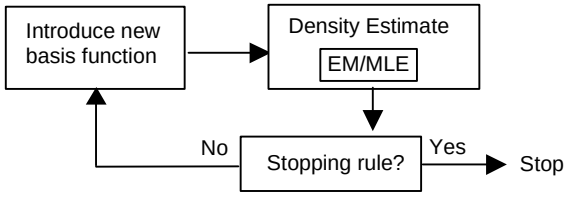

Figure 3. Proposed modeling framework. Basis functions are gradually added into the data representation until the stopping criterion is satisfied.

$$\mathbf{n} \sim N(\mathbf{0}, \sigma_N^2 \cdot I_n) \tag{9}$$

$$\begin{bmatrix} W_N \\ \Theta_N \end{bmatrix} \sim N(\mu_N, \Sigma_N) \tag{10}$$

Denote $\Phi_N = \{\mu_N, \Sigma_N, \sigma_N^2\}$. Equations (8)-(10) define a family of density functions parameterized by Φ_N. Given M independently identical distributed data samples $\{\mathbf{y}_1, \cdots, \mathbf{y}_M\}$, the density estimate $\hat{p}(\mathbf{y})$, in the maximum likelihood (ML) sense, is then defined by the ML estimate of Φ_N such that the likelihood

$$p(\mathbf{y}_1, \cdots, \mathbf{y}_M \mid \Phi_N) = \prod_{i=1}^{i=M} p(\mathbf{y}_i \mid \Phi_N) \tag{11}$$

is maximized over the parameterized family (8),

$$\begin{aligned} \hat{p}(\mathbf{y}) &= p(\mathbf{y} \mid \hat{\Phi}_N) \\ \hat{\Phi}_N &= \underset{\Phi \in \Omega_N}{\operatorname{argmax}} \prod_{i=1}^{M} p(\mathbf{y}_i \mid \Phi) \end{aligned} \tag{12}$$

where Ω_N denotes the domain of Φ_N. Within the framework of ML estimation, $\Phi_N = \{\mu_N, \Sigma_N, \sigma_N^2\}$ defines the *hyper-parameter* set.

To solve (12), we need to answer two questions. First, how many basis functions should be used. We address this issue by introducing a progressive density approximation scheme. Second, given N, how to find basis functions with randomized parameters and solve the ML estimation (12). We answer this question in section 4.

3.2. Progressive density approximation

Involving more basis functions may increase the effective dimension of the model (8)-(10), i.e. the dimensionality of $\mathbf{\Theta}_N$, as well as the computational load. On the other hand, the involvement of more basis functions extends the parametric family of distributions (8) and therefore may increase the accuracy of density estimation (12). Intuitively, N should be large enough to assure the accuracy of density estimation. Meanwhile, N should be bounded to avoid over-fitting.

Before introducing the progressive density approximation method, we first derive a measurement for the accuracy of density estimation. Assume $p_t(\mathbf{y})$ is the true density function for the observed data samples $\{\mathbf{y}_1, \cdots, \mathbf{y}_M\}$, and $\hat{p}(\mathbf{y})$ is the estimated density. The Kullback-Leibler divergence $D(p_t \parallel \hat{p})$ is an effective measure to evaluate the similarity between the two density functions,

$$\begin{aligned} D(p_t \parallel \hat{p}) &= \int p_t(\mathbf{y}) \cdot \log \frac{p_t(\mathbf{y})}{\hat{p}(\mathbf{y})} d\mathbf{y} \\ &= E_{p_t}[\log p_t(\mathbf{y})] - E_{p_t}[\log \hat{p}(\mathbf{y})] \end{aligned} \tag{13}$$

$D(p_t \parallel \hat{p})$ is nonnegative and equal to zero only when the two densities coincide. Since the term $E_{p_t}[\log p_t(\mathbf{y})]$ is independent of the density estimate $\hat{p}(\mathbf{y})$, an equivalent similarity measurement can be defined as

$$\begin{aligned} L(\hat{p}) &= E_{p_t}[\log \hat{p}(\mathbf{y})] \\ &= -D(p_t \parallel \hat{p}) + E_{p_t}[\log p_t(\mathbf{y})] \\ &\le E_{p_t}[\log p_t(\mathbf{y})] \end{aligned} \tag{14}$$

$L(\hat{p})$ increases as estimated density $\hat{p}$ gets closer to the true density p_t. It is upper bounded by $E_{p_t}[\log p_t(\mathbf{y})]$. Since p_t is unknown, $L(\hat{p})$, the expectation of $\log \hat{p}(\mathbf{y})$, can be estimated in practice by its sample mean.

$$\hat{L}(\hat{p}) = \frac{1}{M} \sum_{i=1}^{M} \log \hat{p}(\mathbf{y}_i) \tag{15}$$

Figure 4 illustrates the process of density estimate. By introducing more basis functions, the density estimate gradually approaches the true distribution. The algorithm is summarized below. A similar idea of progressive density approximation is also found in a recent work [11] where the basic components directly modify the density function.

Progressive Density Approximation

Step 1: Start with $N = 1$.

Step 2: Find the ML density estimate $\hat{p}_N$ (12).

Step 3: Increase *N* by 1, and repeat **Step 2** until the stopping rule is satisfied.

Fact: $\hat{p}_N$ gets closer to the true density p_t as N increases.

The algorithm produces a sequence of density estimates $\{\hat{p}_0, \hat{p}_1, \cdots, \hat{p}_N, \cdots\}$,

$$\begin{aligned} &\hat{p}_N(\mathbf{y}) = p(\mathbf{y} \mid \hat{\Phi}_N) \qquad \text{and} \\ &\hat{\Phi}_N = \arg\max_{\Phi_N \in \Omega_N} p(\mathbf{y}_1, \cdots \mathbf{y}_M \mid \Phi_N) \end{aligned} \tag{16}$$

Since the domain Ω_N of the parameter set Φ_N is included into Φ_{N+1},

$$\Omega_N = \{\Phi : \Phi \in \Omega_{N+1}, P(w_{N+1} = 0) = 1\} \subseteq \Omega_{N+1} \tag{17}$$

and $\hat{L}(\hat{p})$ (15) is the exact target function for the ML estimate (12), the sequence $\{\hat{L}(\hat{p}_N)\}$ is increasing monotonically,

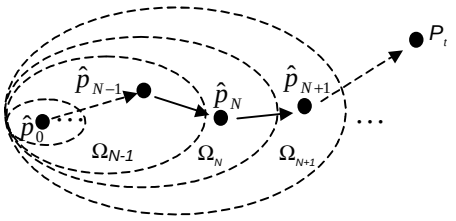

Figure 4. Progressive density approximation. A sequence of density estimate $\hat{p}_N$ progressively approaches the true density function p_t as more basis functions are introduced.

$$\cdots \le \hat{L}(\hat{p}_{N-1}) \le \hat{L}(\hat{p}_N) \le \hat{L}(\hat{p}_{N+1}) \le \cdots \tag{18}$$

This indicates that the sequence of density estimates $\{\hat{p}_N\}$ progressively approaches the true density. To avoid over-fitting, a practical stopping rule is to examine the increasing of $\hat{L}(\hat{p}_N)$ and stop at the point where the increasing of $\hat{L}(\hat{p}_N)$ saturates.

4. Hyper-parameter estimation

The progressive density approximation assures increasing accuracy of the density estimates. For fixed N, we need to solve (12) to get the ML estimate of Φ_N. The expectation-maximization (EM) algorithm is ideally suited to such a problem because of the unknown variable set $\mathbf{\Theta}_N$.

4.1. EM algorithm

In our problem, $\mathbf{\Theta}_N$ is the unknown parameter set, $Y = \{\mathbf{y}_1, \cdots \mathbf{y}_M\}$ is a group of independently observed data, and $\Phi_N = \{\mu_N, \Sigma_N, \sigma_N^2\}$ are the *hyper-parameters* to be estimated. Here μ_N, Σ_N and σ_N^2 denote respectively the mean and covariance of $\mathbf{\Theta}_N$ and the variance of $\mathbf{n}$. The density function for the observed data Y is

$$\begin{aligned} &p(Y \mid \Phi_N) = \prod_{i=1}^{M} p(\mathbf{y}_i \mid \Phi_N), \qquad \text{where} \\ &p(\mathbf{y}_i \mid \Phi_N) = \int p(\mathbf{y}_i \mid \mathbf{\Theta}_{N,i}, \Phi_N)\, p(\mathbf{\Theta}_{N,i} \mid \Phi_N) d\mathbf{\Theta}_{N,i} \end{aligned} \tag{19}$$

EM algorithm

E-step: Compute the expectation of the likelihood

$$\begin{aligned} &Q(\Phi_N \mid \Phi_N^{(k)}) \\ &= \sum_{i=1}^{M} E[\log p(\mathbf{y}_i \mid \mathbf{\Theta}_{N,i}, \Phi_N) + \log p(\mathbf{\Theta}_{N,i} \mid \Phi_N) \mid \mathbf{y}_i, \Phi_N^{(k)}] \\ &= \sum_{i=1}^{M} \int [\log p(\mathbf{y}_i \mid \mathbf{\Theta}_{N,i}, \Phi_N) + \log p(\mathbf{\Theta}_{N,i} \mid \Phi_N)] \\ &\qquad \cdot p(\mathbf{\Theta}_{N,i} \mid \mathbf{y}_i, \Phi_N^{(k)}) d\mathbf{\Theta}_{N,i} \end{aligned} \tag{20}$$

M-step: Maximize the expectation

$$\Phi_N^{(k+1)} = \arg\max_{\Phi_N \in \Omega_N} Q(\Phi_N \mid \Phi_N^{(k)}) \tag{21}$$

The conditional density $p(\boldsymbol{\Theta}_{N,i} \mid \mathbf{y}_i, \Phi_N^{(k)})$ of $\boldsymbol{\Theta}_{N,i}$ given the observed data $\mathbf{y}_i$ and the estimated hyper-parameters $\Phi_N^{(k)}$ from *k*-th iteration is

$$p(\boldsymbol{\Theta}_{N,i} \mid \mathbf{y}_i, \Phi_N^{(k)}) = \frac{p(\boldsymbol{\Theta}_{N,i} \mid \Phi_N^{(k)}) \cdot p(\mathbf{y}_i \mid \boldsymbol{\Theta}_{N,i}, \Phi_N^{(k)})}{p(\mathbf{y}_i \mid \Phi_N^{(k)})} \tag{22}$$

where $p(\mathbf{y}_i \mid \boldsymbol{\Theta}_{N,i}, \Phi_N) = N(\sum_{j=1}^{N} w_{i,j}\mathbf{b}(\theta_{i,j}), \sigma_N^2 \cdot I_n)$

$p(\mathbf{y}_i \mid \boldsymbol{\Theta}_{N,i}, \Phi_N^{(k)}) = N(\sum_{j=1}^{N} w_{i,j}\mathbf{b}(\theta_{i,j}), \sigma_N^{2\,(k)} \cdot I_n)$

$p(\boldsymbol{\Theta}_{N,i} \mid \Phi_N) = N(\mu_N, \Sigma_N)$; $p(\boldsymbol{\Theta}_{N,i} \mid \Phi_N^{(k)}) = N(\mu_N^{(k)}, \Sigma_N^{(k)})$

$\Phi_N^{(k)} = \{\mu_N^{(k)}, \Sigma_N^{(k)}, \sigma_N^{2\,(k)}\}$

$N(\mu, \Sigma)$ denotes multivariate Gaussian density function with mean μ and covariance Σ. EM algorithm starts from an initial guess of Φ_N and proceeds iteratively. Each iteration increases the likelihood function $p(\mathbf{y}_1, \cdots, \mathbf{y}_M \mid \Phi)$ until a global or local maximum is reached.

$$\Phi_N^{(0)} \xrightarrow{E-step} Q(\Phi_N \mid \Phi_N^{(0)}) \xrightarrow{M-step} \Phi_N^{(1)} \rightarrow \cdots \rightarrow \hat{\Phi}_N$$

4.2. Practical implementation

When the nonlinear basis function $\mathbf{b}(\theta)$ is involved in the integrand in (20), the computation of function *Q* is not always tractable. We propose a suboptimal solution to estimate Φ_N. $Q(\Phi_N \mid \Phi_N^{(k)})$ is defined as the expectation of a function of $\boldsymbol{\Theta}_N$ (20), and we approximate the integration by the function value at the point of the ML estimate of $\boldsymbol{\Theta}_N$.

$$Q(\Phi_N \mid \Phi_N^{(k)}) \cong \sum_{i=1}^{M} [\log p(\mathbf{y}_i \mid \hat{\boldsymbol{\Theta}}_{N,i}^{(k)}, \Phi_N) + \log p(\hat{\boldsymbol{\Theta}}_{N,i}^{(k)} \mid \Phi_N)]$$

The approximation is accurate when the distribution of $\boldsymbol{\Theta}_N$ is well concentrated around its mean. Thus, the EM algorithm reduces to the iterative ML estimation of $\boldsymbol{\Theta}_N$ and Φ_N.

$$\Phi_N^{(0)} \xrightarrow{MLE} \hat{\boldsymbol{\Theta}}_N^{(0)} \xrightarrow{MLE} \Phi_N^{(1)} \xrightarrow{MLE} \hat{\boldsymbol{\Theta}}_N^{(1)} \rightarrow \cdots \rightarrow \hat{\Phi}_N \rightarrow \hat{\boldsymbol{\Theta}}_N$$

Each iteration increases the likelihood function $p(\mathbf{y}_1, \cdots, \mathbf{y}_M; \boldsymbol{\Theta}_{N,1}, \cdots, \boldsymbol{\Theta}_{N,M} \mid \Phi)$ until a global or local maximum is reached.

Suboptimal solution

MLE of $\boldsymbol{\Theta}_N$:

$$\hat{\boldsymbol{\Theta}}_{N,i}^{(k)} = \arg\max_{\boldsymbol{\Theta}_N} p(\boldsymbol{\Theta}_N \mid \Phi_N^{(k)}) \cdot p(\mathbf{y}_i \mid \boldsymbol{\Theta}_N, \Phi_N^{(k)}) \quad (i = 1, \cdots, M) \tag{23}$$

MLE of Φ_N:

$$\Phi_N^{(k+1)} = \arg\max_{\Phi_N} \sum_{i=1}^{M} [\log p(\mathbf{y}_i \mid \hat{\boldsymbol{\Theta}}_{N,i}^{(k)}, \Phi_N) + \log p(\hat{\boldsymbol{\Theta}}_{N,i}^{(k)} \mid \Phi_N)] \tag{24}$$

The ML estimate of $\boldsymbol{\Theta}_N$ (23) can be solved through nonlinear optimization. When the density functions involved in (24) are Gaussian, the update rules for Φ_N are

$$\begin{aligned} \mu_N^{(k+1)} &= \frac{1}{M}\sum_{i=1}^{M} \boldsymbol{\Theta}_{N,i}^{(k)} \\ \Sigma_N^{(k+1)} &= \frac{1}{M}\sum_{i=1}^{M} [\boldsymbol{\Theta}_{N,i}^{(k)} - \mu_N^{(k+1)}][\boldsymbol{\Theta}_{N,i}^{(k)} - \mu_N^{(k+1)}]^T \\ \sigma_N^{2\,(k+1)} &= \frac{1}{Mn}\sum_{i=1}^{M} \| \mathbf{y}_i - \sum_{j=1}^{N} w_{i,j}\mathbf{b}(\theta_{i,j}) \|^2 \end{aligned} \tag{25}$$

As the number of basis functions *N* increases, the random vector $\boldsymbol{\Theta}_N$ is likely to belong to a lower dimensional space, i.e. its covariance matrix Σ_N is singular and we can no longer write out the density function $p(\boldsymbol{\Theta}_N \mid \Phi_N)$ used in (19), (23)-(24). In such cases, we reduce the dimension of $\boldsymbol{\Theta}_N$ through the linear transformation $\boldsymbol{\beta} = A^T \cdot \boldsymbol{\Theta}_N$; $\boldsymbol{\Theta}_N = A \cdot \boldsymbol{\beta}$, where A is composed of the eigenvectors of Σ_N corresponding to the nonzero eigenvalues, and the covariance matrix of $\boldsymbol{\beta}$ has full rank. The density approximation and parameter estimation are actually performed on $\boldsymbol{\beta}$.

4.3. Adding a new basis function

The estimate $\hat{\Phi}_N = \{\hat{\mu}_N, \hat{\Sigma}_N, \hat{\sigma}_N^2\}$ is used to initialize $\Phi_{N+1} = \{\mu_{N+1}, \Sigma_{N+1}, \sigma_{N+1}^2\}$. Assume that $\{w_{N+1}, \theta_{N+1}\}$ are the parameters for the newly introduced basis function. One choice is to initially assume independence between $\{w_{N+1}, \theta_{N+1}\}$ and $\boldsymbol{\Theta}_N$, and the initialization for Φ_{N+1} becomes

$$\mu_{N+1}^{(0)} = \begin{bmatrix} \hat{\mu}_N \\ \mu_{N+1,0} \end{bmatrix}; \Sigma_{N+1}^{(0)} = \begin{bmatrix} \hat{\Sigma}_N & \mathbf{0} \\ \mathbf{0} & c \cdot I \end{bmatrix}; \sigma_{N+1}^{2\,(0)} = \hat{\sigma}_N^2 \tag{26}$$

where c is an arbitrary positive value. $\mu_{N+1,0} = E[[w_{N+1}, \theta_{N+1}]^T]$ is the initial parameter mean of the unknown $(N+1)$-th basis function. Initialization of $\mu_{N+1,0}$ essentially requires searching for the new basis function that best approximates the multivariate data.

$$\begin{aligned} \mu_{N+1,0} &= \begin{bmatrix} \hat{w}_{N+1} \\ \hat{\theta}_{N+1} \end{bmatrix} = \arg\min_{[w,\theta]^T} \sum_{i=1}^{M} \| \mathbf{r}_i - w\mathbf{b}(\theta) \|^2 \\ \mathbf{r}_i &= \mathbf{y}_i - \sum_{j=1}^{N} \hat{w}_{i,j}\mathbf{b}(\hat{\theta}_{i,j}) \quad (i = 1, \cdots, M) \end{aligned} \tag{27}$$

where $\hat{w}_{i,j}$ and $\hat{\theta}_{i,j}$ are elements of $\hat{\boldsymbol{\Theta}}_N$.

5. Experimental results

5.1. Modeling 1D curves

In this academic example, we want to learn the data distribution from $M = 100$ synthesized samples. A set of M curves $\{f_1(t), \cdots f_M(t)\}$ are obtained from a reference curve $f_0(t)$ by random translation d, scaling s and amplification w, and by adding a white noise term n, $f_i(t) = w_i f_0((t - d_i)/s_i) + n$. Figure 5(a) shows 10 curve realizations. Every curve $f_i(t)$ is sampled at $n = 50$ common locations $\{t_1, \cdots, t_n\}$ and generates a length-n vector $\mathbf{y}_i = [f_i(t_1), \cdots f_i(t_n)]^T$ as observed data. These vector samples spread into a cluster of points in R^n with intrinsic dimension 3. For better visual effect, we plot the distribution manifold by projecting it onto the plane spanned by the first two linear principal components since 80% of the data variance is concentrated in this subspace. The number of data samples M may be small compared to the vector length n, but is large compared to the intrinsic dimension. However, the information of intrinsic dimension as well as curve parameterization is unknown to our modeling scheme.

The progressive density approximation was applied to modeling the data. The parameterized family of DOG wavelets defined in (6) is adopted as basis functions. Each time a new basis function is added on, the MLE algorithm is carried out to obtain the best density estimation $\hat{p}_N$. This process continues until no strong increasing is found in the similarity measure $\hat{L}(\hat{p}_N)$ (15). Figure 5(c)-(f) shows the density estimate in each step as N increases from 1 to 4. With 3 basis functions, the estimated density is already close to the true distribution.

A similar example with only parameterized translation involved is discussed in [6]. Compared to the work in [6] which intends to find the one-dimensional principal manifold, we obtain a density estimate of the data. By performing PCA on the parameter set $\mathbf{\Theta}_N$, we actually perform nonlinear PCA(NLPCA) on the original data. Thus, one-dimensional PCA approximation of $\mathbf{\Theta}_N$ gives the one-dimensional principal manifold of the data distribution.

Figure 5(g) compares the mean curves obtained by linear PCA and NLPCA with the true signal mean, where the nonlinear mean is produced by the mean of $\mathbf{\Theta}_N$. While the linear PCA produces a blurred mean curve, the nonlinear mean curve better approximates the true mean, which is generated by the true parameter mean. Figure 5(h) shows that the increasing of the similarity measure $\hat{L}(\hat{p}_N)$ begins to saturate after 3 basis functions are introduced. Figure 5(i) illustrates one-dimensional linear and nonlinear PCA approximation of the distribution. The nonlinear principal manifold of dimension 2 is also shown in Figure 5(j). N is set to 4 in Figure 5(g), (i) and (j).

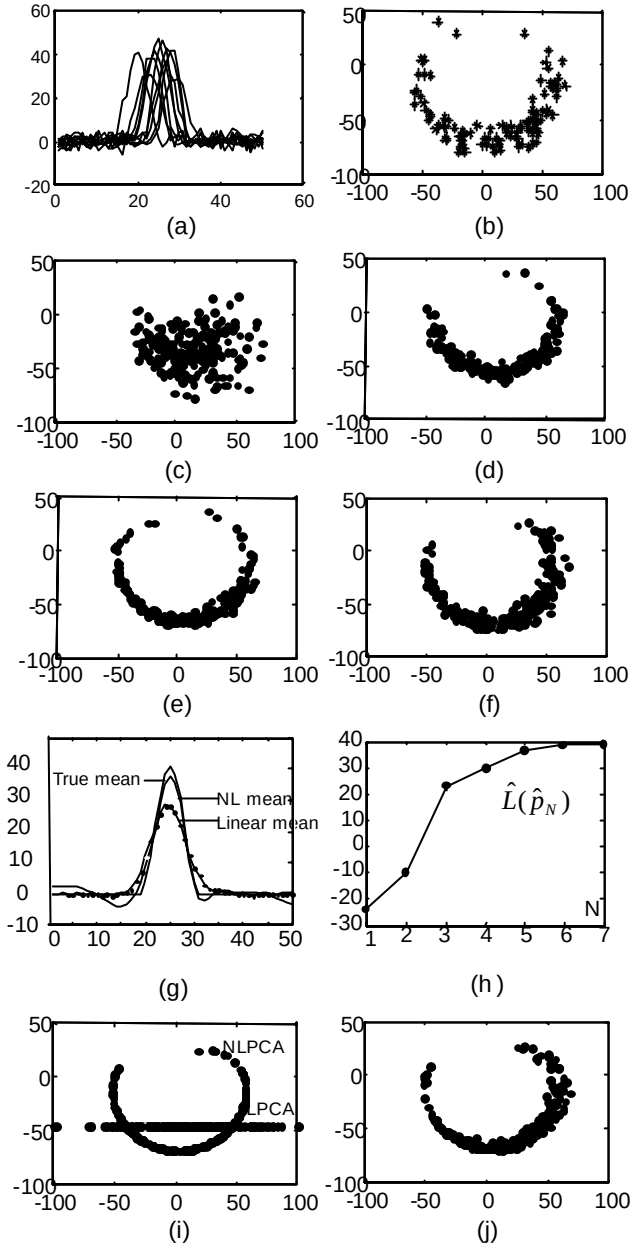

Figure 5. Modeling curves. (a) Curve samples. (b) 2D visualization of data distribution. (c)-(f) Progressive density estimate (N=1,2,3,4). (g) Linear, nonlinear and true mean curves. (h) Density similarity measurement (up to a constant value). (i) Linear PCA and nonlinear principal manifold approximation with dimension 1. (j) Nonlinear manifold approximation with dimension 2. (N=4)

5.2. Modeling object pose

In this experiment, we are interested in modeling the nonlinear manifold formed by the object appearances under varying poses. Images used in this experiment are obtained from the "Columbia Object Image Library", where each object was rotated through 360 degrees and 72 images were

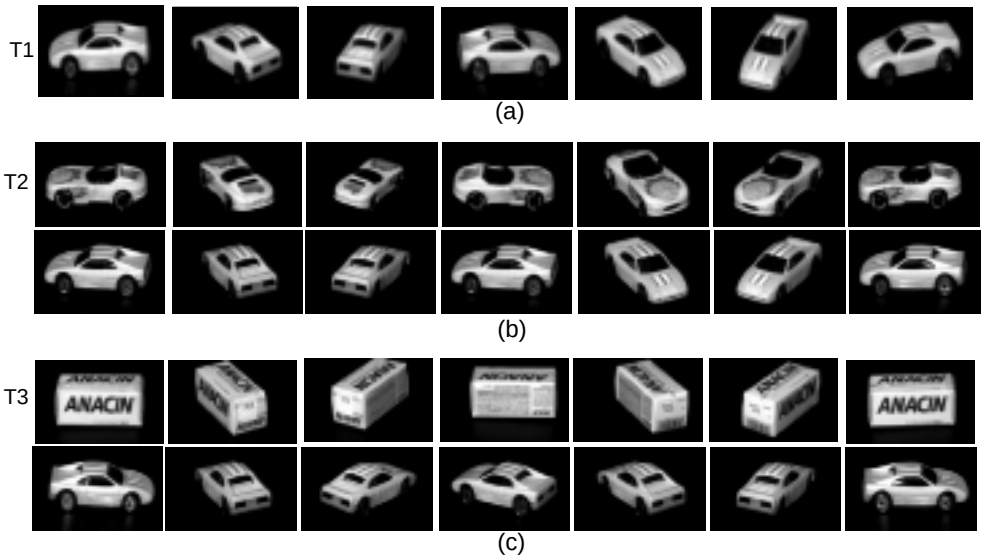

Figure 6. Modeling object pose. (a) Images used for learning. (b) Test object 2. (c) Test object 3. (b)-(c) top row: test images; bottom row: pose estimate results shown by training images.

taken per object, one at every 5 degrees of rotation. Figure 6(a) shows a few images of the object used for modeling. Murase and Nayar have shown in their work [13] the nonlinear shape of the data distribution. They represent the manifold by interpolating data points in the low-dimensional linear subspace.

In our experiment, we use 36 images for learning, one at every 10 degrees of rotation. The remaining ones are left for evaluating the model through pose estimation. Using the prior knowledge that the internal variable $\mathbf{x}$ (pose) is better modeled as uniformly distributed in the range from 0 to 360 degrees, the linearization in (7) would not hold well and higher order terms of $\mathbf{x}$ are required. For this example, we adopt the uniform distribution of $\mathbf{x}$ and the linearization in (7) is replaced by a piecewise cubic polynomial function of $\mathbf{x}$. Thus the random variables $\{W_N, \Theta_N\}$ in (8) are redefined as functions of uniformly distributed random variable $\mathbf{x}$. The polynomial coefficients are included in the hyper-parameter set Φ_N. Gabor wavelets [14], parameterized by their locations, scales and orientations, are employed as our basis functions. Figure 7(a) and (b) visualize respectively, the true data distribution and the distribution learnt through our modeling scheme, where we project data points into the principal subspace of dimension 3 and produce the 3D plot.

Compared with the representation by Murase and Nayar, our representation is associated by a density function. The estimated density imposes discrimination power on the model, which is demonstrated in the experiment of pose estimation. Given an input image $\mathbf{y}$, the pose estimate $\hat{\mathbf{x}}$ under this model is solved by the ML estimation instead of the least Euclidean distance,

$$\begin{aligned}\hat{\mathbf{x}} &= \arg\max_{\mathbf{x}} p(\mathbf{x} \mid \mathbf{y}, \hat{\Phi}) \\ &= \arg\max_{\mathbf{x}} p(\mathbf{y} \mid \mathbf{x}, \hat{\Phi}) \cdot p(\mathbf{x} \mid \hat{\Phi})\end{aligned} \tag{28}$$

The model obtained from the objects shown in Figure 6(a) was used for pose estimate on three sets of images. The first test set (T1) consists of 72 images from the same object. Figure 7(c) shows its results. The average error in the estimated pose value is 1.34 degrees. The second test set (T2) consists of 72 images of a different object. In Figure 6(b), the top row shows a few examples of T2 images and

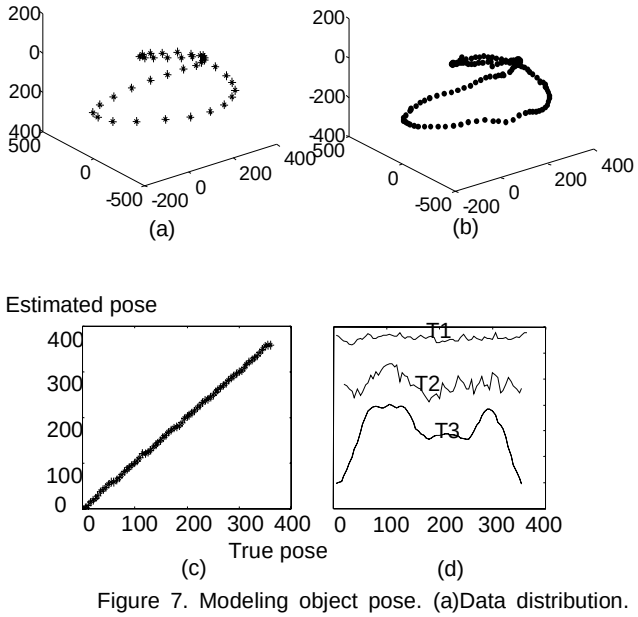

Figure 7. Modeling object pose. (a)Data distribution. (b) Estimated density. (c) Pose estimation result of T1. (d) Likelihood comparison of pose estimation T1-T3

the bottom row shows the estimated poses using the corresponding T1 images. The third test set (T3) is shown in Figure 6(c). In Figure 7(d), we plot the likelihood associated with the pose estimate. Our model clearly indicates that T2 object bears more resemblance to T1 object than T3 object does. This result demonstrates the discrimination ability of our model due to the density approximation. The likelihood provided by the density estimate is a powerful tool for pattern identification.

6. Discussions

This paper presented a general modeling scheme for the statistical characterization of internally unimodal data. The nonlinear structure of the data distribution is revealed by the ML based procedure. Although the experiments included here were only designed to demonstrate the effectiveness of the scheme, the density estimate and the likelihood measure provided by our model are powerful statistical tools for a wider range of vision applications such as object identification and tracking.

Our analysis addresses two important issues in statistical modeling: efficiency and accuracy. Such idea is pursued in the design of the progressive density approximation. The algorithm finds, in a progressive fashion, the set of basis functions that are the most efficient in term of modeling accuracy. Through parametric function representation, we characterize the intrinsic data information by jointly modeling the linear and nonlinear parameter sets. If the nonlinear parameters are deterministic, our scheme simply becomes a linear approach that models the linear coefficients with a fixed set of basis. However, by including the nonlinear parameters into the model, higher order statistics are automatically included. Hence, we can use simple distributions for the parameters, such as the unimodal Gaussian, to describe more complex distributions of the data.

The type of basis and parameters used in the representation is determined by the nature of the data. The wavelet basis is chosen in this work due to its ability of locally decorrelating image data. Nevertheless, the choice of basis and parameter type for a specific application is by itself a research issue.

References

[1] B. Moghaddam and A. Pentland, "Probabilistic Visual Learning for Object Representation", *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, Jul. 1997, pp. 696-710.

[2] M. Tipping and C. Bishop. "Probabilistic Principal Component Analysis". Technical Report NCRG/97/010, Neural Computing Research Group, Aston University, September 1997.

[3] R. A. Johnson and D. W. Wichern, *Applied Multivariate Statistical Analysis*, Prentice Hall, NJ, 1992.

[4] J. Ng and S. Gong, "Multi-view Face Detection and Pose Estimation Using A Composite Support Vector Machine Across the View Sphere", *RATFG-RTS*, 1999, pp. 14-21.

[5] N. Kambhatla and T. K. Leen, "Dimension Reduction by Local PCA", *Neural Computation*, vol. 9, no. 7, Oct. 1997, pp. 1493-516.

[6] B. Chalmond and S. C. Girard, "Nonlinear Modeling of Scattered Multivariate Data and Its Application to Shape Change", *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 21, no. 5, May 1999, pp. 422-434.

[7] B. Moghaddam, "Principal Manifolds and Bayesian Sub-spaces for Visual Recognition", *IEEE Int. Conf. on Computer Vision*, 1999, pp. 1131-1136.

[8] J. O. Ramsay and X. Li, "Curve Registration", *J. R. Statist. Soc., Series B*, vol. 60, 1998, pp. 351-363.

[9] G. James and T. Hastie, "Principal Component Models for Sparse Functional Data", Technical Report, Department of Statistics, Stanford University, 1999.

[10] M. Black, and Y. Yacoob, "Tracking and Recognizing Rigid and Non-Rigid Facial Motions Using Local Parametric Models of Image Motion", *IEEE Int. Conf. Computer Vision*, 1995, pp. 374-381.

[11] Z. R. Yang and M. Zwolinski, "Mutual Information Theory for Adaptive Mixture Models", *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 23, no. 4, Apr. 2001, pp. 396-403.

[12] A. P. Dempster, N. M. Laird and D. B. Rubin, "Maximum Likelihood from Incomplete Data via EM Algorithm", *J. R. Statist. Soc., Series B*, vol. 39, 1977, pp. 1-38.

[13] H. Murase and S. K. Nayar, "Visual Learning and Recognition of 3-D Objects from Appearance", *Int. J. Computer Vision*, vol. 14, 1995, pp. 5-24.

[14] Q. Zhang and A. Benveniste, "Wavelet Networks", *IEEE Trans. Neural Networks*, vol. 3, no. 6, Nov 1992, pp. 889-898.

[15] C. M. Bishop and J. M. Winn, "Non-linear Bayesian Image Modelling", *European Conf. on Computer Vision*, 2000, pp. 3-17.

[16] B. Frey and N. Jojic, "Transformed Component Analysis: Joint Estimation of Spatial Transformations and image Components", *IEEE Int, Conf. Computer Vision*, 1999, pp. 1190-1196.

[17] M. Weber, M. Welling and P. Perona, "Unsupervised Learning of Models for Recognition", *European Conf. on Computer Vision*, 2000, pp. 18-32.

[18] T. S. Lee, "Image Representation Using 2D Gabor Wavelets", *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 18, no. 10, 1996, pp. 959-971.

[19] B.W. Silverman, "Incorporating Parametric Effects into Functional Principal Components Analysis, *J. R. Statist. Soc., Series B*, vol. 57, no. 4, 1995, pp. 673-689.

[20] M. Black, and A. Jepson, "Eigentracking: Robust Matching and Tracking of Articulated Objects Using A View-based Representation", *ECCV*, 1996, pp. 329-342.

[21] A. R. Gallant, *Nonlinear Statistical Models*, John Wiley & Sons Inc., NY, 1987.

[22] J. B. Tenenbaum, V. De Silva, and J. C. Langford, "A Global Geometric Framework for Nonlinear Dimensionality Reduction", *Science*, vol. 290, 2000, pp. 2319-2323.

[23] S. Roweis and L. Saul, "Nonlinear Dimensionality Reduction by Locally Linear Embedding", *Science*, vol. 290, 2000, pp.2323-2326.