Sparse density estimation on the multinomial manifold

Article

Accepted Version

Sparse Density Estimation on the Multinomial Manifold

Xia Hong, Junbin Gao, Sheng Chen, and Tanveer Zia

***Abstract*— A new sparse kernel density estimator is introduced based on the minimum integrated square error criterion for the finite mixture model. Since the constraint on the mixing coefficients of the finite mixture model is on the multinomial manifold, we use the well-known Riemannian trust-region (RTR) algorithm for solving this problem. The first- and second-order Riemannian geometry of the multinomial manifold are derived and utilized in the RTR algorithm. Numerical examples are employed to demonstrate that the proposed approach is effective in constructing sparse kernel density estimators with an accuracy competitive with those of existing kernel density estimators.**

***Index Terms*— Minimum integrated square error (MISE), multinomial manifold, probability density function (pdf), sparse modeling.**

I. Introduction

The probability density function (pdf) estimation problem is fundamental to many data analysis and pattern recognition applications [1]–[6]. The identification of the finite mixture model is usually based on the expectation-maximization (EM) algorithm [7], which provides the maximum likelihood (ML) estimator of the mixture model's parameters, while the number of mixtures is preset. This associated ML optimization is generally a highly nonlinear optimization process requiring extensive computation. While the EM algorithm for Gaussian mixture model enjoys an explicit iterative form [8], it is also known that this EM algorithm-based ML estimation has a low convergence speed. To tackle the associated numerical difficulties, it is often required to apply resampling techniques [9], [10]. Alternatively, the Parzen window (PW) estimator [11] can be regarded as a special case of the finite mixture model [1], in which the number of mixtures is equal to that of the training data samples and all the mixing weights are equal. The point density estimate using the PW estimator for a future data sample can be computationally expensive if the number of training data samples is very large.

There is a considerable interest in research on sparse pdf estimation. The support vector machine (SVM) density estimation technique has been proposed in [12] and [13]. The optimization in the SVM method is to solve a constrained quadratic optimization problem. This yields the sparsity-inducing property, i.e., at the optimality, many kernels' weights are driven to zeros. Alternatively, a novel regression-based pdf estimation method has been introduced [14], in which the empirical cumulative distribution function (cdf) is constructed, in the same manner as in the SVM density estimation approach, to be used as the desired response. The orthogonal forward regression (OFR) approach is an efficient supervised regression model construction method [15]. The OFR method has been combined with a leave-one-out test score and local regularization [16], [17]. The regression-based idea of [14] and the approach in [16] and [17] have been extended to yield a new OFR-based sparse density estimation algorithm [18] with a performance comparable with that of the PW estimator. In [14] and [18], the regressors are the cdfs of the kernels and the target response is the empirical cdf. A simple and viable alternative approach has been proposed to use kernels directly as regressors by adopting the PW estimator as the target response [19].

The desirable property of sparsity inducing also happens in the interesting approach of reduced set density estimator (RSDE) [20], based on the minimization of the integrated square error between the estimator and the true density [2], [20], [21] and they introduced two efficient optimization algorithms. Our extensive experience has shown that all the sparse density estimators [12], [13], [18]–[20] discussed here are capable of automatically producing sparse pdf estimates with a performance comparable with that of the PW estimator, but the density estimators of [18]–[20] produce much sparser estimates than the SVM-based density estimator. Recently, a new sparse kernel density estimator has been introduced for sparse kernel density estimation with very low computational cost, based on the MISE and the forward constrained regression (FCR) [22]. In [23] a recursive algorithm, referred to as the FCR-MISE algorithm, has been proposed for the selection of significant kernels one at time using the minimum integrated square error (MISE) criterion for both kernel selection and the estimation of mixing weights.

Recent years have witnessed great development in Riemannian optimization algorithms on many types of matrix manifolds, such as the Stiefel manifold, Grassmann manifold, and the manifold of positive definite matrices [24, Sec. 3.4]. Since Riemannian optimization is directly based on the curved manifolds, one can eliminate those constraints such as orthogonality to obtain an unconstrained optimization problem that, by construction, will use only feasible points. This allows one to incorporate Riemannian geometry in the resulting optimization problems, thus producing far more accurate numerical results. The recent successful applications of Riemannian optimization in machine learning, computer vision, and data mining, citing a few, include fixed low-rank optimization [25], Riemannian dictionary learning [26], and computer vision tasks [27].

Against this background, this brief introduces a new sparse kernel density estimator based on the MISE criterion for the finite mixture model. Recognizing that the constraint on the mixing coefficients of the finite mixture model is the multinomial manifold, the well-known Riemannian trust-region (RTR) algorithm can be readily used for solving this problem. Clearly, the first- and second-order Riemannian geometry of the multinomial manifold is required and this is developed in this contribution. The proposed algorithm is referred to as the RTR-MISE algorithm. Numerical examples are employed to demonstrate that RTR-MISE is effective in constructing sparse kernel density estimators with an accuracy competitive with those of existing kernel density estimators.

Manuscript received July 2, 2014; revised January 3, 2015; accepted January 4, 2015. This work was supported by the Australian Research Council through the Discovery Project under Grant DP130100364.

X. Hong is with the School of Systems Engineering, University of Reading, Reading RG6 6AY, U.K. (e-mail: x.hong@reading.ac.uk).

J. Gao and T. Zia are with the School of Computing and Mathematics, Charles Sturt University, Bathurst, NSW 2795, Australia (e-mail: jbgao@csu.edu.au; tzia@csu.edu.au).

S. Chen is with the Department of Electronics and Computer Science, University of Southampton, Southampton SO17 1BJ, U.K., and also with the Faculty of Engineering, King Abdulaziz University, Jeddah 21589, Saudi Arabia (e-mail: sqc@ecs.soton.ac.uk).

Digital Object Identifier 10.1109/TNNLS.2015.2389273

II. Sparse Density Estimation Using Minimal Integrated Square Error

Given a finite data set $D_N = \{\boldsymbol{x}_j\}_{j=1}^N$ consisting of N data samples, where the data vector $\boldsymbol{x}_j \in \mathbb{R}^m$ follows an unknown pdf $p(\boldsymbol{x})$, the problem under study is to find a sparse approximation of $p(\boldsymbol{x})$ based on D_N. A general kernel-based density estimator of $p(\boldsymbol{x})$ is given by

$$\hat{p}(\boldsymbol{x};\boldsymbol{\beta},\rho) = \sum_{j=1}^{N} \beta_j K_\rho(\boldsymbol{x},\boldsymbol{x}_j) \tag{1}$$

s.t.

$$\beta_j \geq 0, \quad 1 \leq j \leq N, \quad \text{and } \boldsymbol{\beta}^T\mathbf{1} = 1 \tag{2}$$

where β_j's are the kernel weights, $\boldsymbol{\beta} = [\beta_1, \beta_2, \ldots, \beta_N]^T$, and $\mathbf{1}$ is the vector whose elements are all equal to one, while $K_\rho(\boldsymbol{x},\boldsymbol{x}_j)$ is a chosen kernel function with the kernel centre vector $\boldsymbol{x}_j$ and a suitable kernel width ρ. In this brief, we use the Gaussian kernel of

$$K_\rho(\boldsymbol{x},\boldsymbol{x}_j) = \frac{1}{(2\pi\rho^2)^{m/2}} \exp\left(-\frac{\|\boldsymbol{x}-\boldsymbol{x}_j\|^2}{2\rho^2}\right) \tag{3}$$

but many other kernels can also be used. The well-known PW estimator, denoted by $\hat{p}^{\text{Par}}(\mathbf{x})$, is a special case of (1) with $\beta_j = (1/N)$, $\forall j$.

The log-likelihood for $\boldsymbol{\beta}$ can be formed using observed data D_N, denoted by $\log L$, as

$$\begin{aligned}\log L &= \frac{1}{N}\sum_{i=1}^{N} \log \hat{p}(\boldsymbol{x}_i;\boldsymbol{\beta},\rho) \\ &= \frac{1}{N}\sum_{i=1}^{N} \log\left(\sum_{j=1}^{N} \beta_j K_\rho(\boldsymbol{x}_i,\boldsymbol{x}_j)\right).\end{aligned} \tag{4}$$

Note that by the law of large numbers, the log-likelihood of (4) tends to

$$\int_{\Re^m} p(\boldsymbol{x}) \log \hat{p}(\boldsymbol{x};\boldsymbol{\beta},\rho) d\boldsymbol{x} \tag{5}$$

as $N \to \infty$ with probability one. Equation (4) is simply the negative cross-entropy or divergence between the true density $p(\boldsymbol{x})$ and the estimator $\hat{p}(\boldsymbol{x};\boldsymbol{\beta},\rho)$. It can be shown that the PW estimator $\beta_j^{\text{Par}} = (1/N)$, $\forall j$ can be obtained as an optimal estimator via the maximization of (4) with respective to $\boldsymbol{\beta}$ subject to the constraints $\beta_j \geq 0$, $j = 1, \ldots, N$, $\boldsymbol{\beta}^T\mathbf{1} = 1$.

The MISE between a pdf estimator and the true density is a classical goodness of fit criterion of probability density estimation, both for nonparametric [2], [20] and for parametric models [21]. The argument $\hat{\boldsymbol{\beta}}$ which provides the MISE is given by

$$\begin{aligned}\hat{\boldsymbol{\beta}} &= \text{argmin}_{\boldsymbol{\beta}} \int (p(\boldsymbol{x}) - \hat{p}(\boldsymbol{x};\boldsymbol{\beta}_N,\rho))^2 d\boldsymbol{x} \\ &= \text{argmin}_{\boldsymbol{\beta}} \left\{\int \hat{p}^2(\boldsymbol{x};\boldsymbol{\beta}_N,\rho)d\boldsymbol{x} - 2E[\hat{p}(\boldsymbol{x};\boldsymbol{\beta}_N,\rho)]\right\}\end{aligned} \tag{6}$$

where the term $\int (p(\boldsymbol{x}))^2 d\boldsymbol{x}$ has been dropped from the cost function due to its independence to $\boldsymbol{\beta}$. $E[\bullet]$ denotes the expectation with respect to the true density $p(\boldsymbol{x})$. Substituting (1) into (6), we have the MISE estimator given as

$$\begin{aligned}\hat{\boldsymbol{\beta}} &= \text{argmin}_{\boldsymbol{\beta}} \left\{\sum_{i=1}^{N}\sum_{j=1}^{N} \beta_i\beta_j \int K_\rho(\boldsymbol{x},\boldsymbol{x}_i)K_\rho(\boldsymbol{x},\boldsymbol{x}_j)d\boldsymbol{x} \right. \\ &\qquad \left. -2\sum_{j=1}^{N} \beta_j E[K_\rho(\boldsymbol{x},\boldsymbol{x}_j)]\right\} \\ &= \text{argmin}_{\boldsymbol{\beta}} \left\{\frac{1}{2}\boldsymbol{\beta}^T\boldsymbol{Q}\boldsymbol{\beta} - \boldsymbol{\beta}^T\boldsymbol{p}\right\}\end{aligned} \tag{7}$$

where $\boldsymbol{Q} = \{q_{i,j}\} \in \Re^{N\times N}$ is a matrix with its elements $q_{i,j}$ as $K_{\sqrt{2}\rho}(\boldsymbol{x}_i,\boldsymbol{x}_j)$ and $\boldsymbol{p} = [\hat{p}^{\text{Par}}(\boldsymbol{x}_1), \ldots, \hat{p}^{\text{Par}}(\boldsymbol{x}_N)]^T \in \Re^{N\times 1}$ is a vector with the elements of $\hat{p}^{\text{Par}}(\boldsymbol{x}_j) = (1/N)\sum_{i=1}^{N} K_\rho(\boldsymbol{x}_i,\boldsymbol{x}_j)$, which is the Parzen pdf estimation for point $\boldsymbol{x}_j$ using kernel (3). Note that in (7), the identity $\int K_\rho(\boldsymbol{x},\boldsymbol{x}_i)K_\rho(\boldsymbol{x},\boldsymbol{x}_j)d\boldsymbol{x} = K_{\sqrt{2}\rho}(\boldsymbol{x}_i,\boldsymbol{x}_j)$ was applied.

As discussed in [2] and [20], the above MISE estimator over the constraint (2) will end up setting many $\beta_j's$ associated with low density $\hat{p}^{\text{Par}}(\boldsymbol{x}_j)$ as zeros. However, the resultant estimator may still not be very sparse. In linear-in-the-parameters modeling and kernel methods, the number of terms in the model is referred as the l_0-norm of the parameter vector. Minimizing such quantity is related to variable and feature selection, ensuring model sparsity and generalization [28]. Because of the intractability in the minimization of the l_0-norm, there is considerable research on the approximation schemes on the l_0-norm [28] and the associated computational complexities. It was analyzed that when combined with the convexity constraint of the kernel parameter vector, model sparsity can be achieved by maximizing of the l_2-norm of the parameters [29]. Thus, our optimization problem is given as

$$\min_{\boldsymbol{\beta}\in\mathbb{P}^N} \quad \left\{F(\boldsymbol{\beta}) = \frac{1}{2}\boldsymbol{\beta}^T\boldsymbol{C}\boldsymbol{\beta} - \boldsymbol{\beta}^T\boldsymbol{p}\right\} \tag{8}$$

$$\text{s.t.} \qquad \beta_j \geq 0, \ 1 \leq j \leq N, \ \text{and } \boldsymbol{\beta}^T\mathbf{1} = 1 \tag{9}$$

where $\boldsymbol{C} = \boldsymbol{Q} - \delta\boldsymbol{I}$, $\boldsymbol{I}$ is the identity matrix. δ is zero or a very small preset positive number that directly controls the sparsity. The larger the value δ, the sparser the model.

III. Multinomial Manifold

In this section, we briefly introduce the concept of multinomial manifold and the necessary ingredients used in the retraction-based framework of Riemannian optimization. The main notations on Riemmann geometry on multinomial manifold in this section is summarized in Table I as a reference. We refer the readers to [24] for the general concepts of manifolds.

The multinomial manifold (also called a simplex) is the parameter space of the multinomial distribution defined by

$$\begin{aligned}\mathbb{P}^N = \big\{\boldsymbol{\beta} = [\beta_1, \ldots, \beta_N]^T \in \Re^N : \\ \beta_j > 0, \ 1 \leq j \leq N, \boldsymbol{\beta}^T\mathbf{1} = 1\big\}.\end{aligned} \tag{10}$$

The multinomial manifold $\mathbb{P}^N$ is an embedded Riemannian submanifold of the Euclidean space $\Re^N$ equipped with the so-called Fisher information metric $g(\boldsymbol{u}_{\boldsymbol{\beta}}, \boldsymbol{v}_{\boldsymbol{\beta}})$ as [30], [31]

$$g(\boldsymbol{u}_{\boldsymbol{\beta}}, \boldsymbol{v}_{\boldsymbol{\beta}}) = \sum_{i=1}^{N} \frac{u_i v_i}{\beta_i} \tag{11}$$

where $\boldsymbol{u}_{\boldsymbol{\beta}}, \boldsymbol{v}_{\boldsymbol{\beta}} \in T_{\boldsymbol{\beta}}\mathbb{P}^N \subset \Re^N$ are tangent vectors to $\mathbb{P}^N$ at $\boldsymbol{\beta}$, with their ith elements defined as u_i and v_i, respectively. The inner product on $\mathbb{P}^N$ as defined in (11) determines the geometry, such as

TABLE I
NOTATIONS FOR MULTINOMIAL MANIFOLD

$\{\mathbb{P}^N, g\}$	Multinomial manifold for parameter vector $\boldsymbol{\beta}$ and the inner product of the manifold
$T_{\boldsymbol{\beta}}\mathbb{P}^N$	Tangent space of the multinomial manifold
$\boldsymbol{u}_{\boldsymbol{\beta}}, \boldsymbol{v}_{\boldsymbol{\beta}}$	Tangent vectors at $\boldsymbol{\beta}$
$\text{Proj}_{\boldsymbol{\beta}}(\boldsymbol{z})$	Orthogonal projector from a vector in ambient space to the tangent space at $\boldsymbol{\beta}$
$\text{grad}F(\boldsymbol{\beta})$	Riemannian gradient of $F(\boldsymbol{\beta})$ on the manifold $\mathbb{P}^N$
$\text{Grad}F(\boldsymbol{\beta})$	The classical gradient of $F(\boldsymbol{\beta})$ as seen in Euclidean space
$D\boldsymbol{f}(\boldsymbol{x})$	Euclidean gradient directional derivative of $\boldsymbol{f}$ in the direction of $\boldsymbol{x}$
$\nabla_{\boldsymbol{u}}\boldsymbol{v}$	Riemannian connection on the manifold $\mathbb{P}^N$
$\overline{\nabla}_{\boldsymbol{u}}\boldsymbol{v}$	The connection on Euclidean space
$R_{\boldsymbol{\beta}}(\cdot)$	Retraction mapping

distance, angle, and curvature on $\mathbb{P}^N$. Note that the tangent space $T_{\boldsymbol{\beta}}\mathbb{P}^N$ at element $\boldsymbol{\beta}$ can be described by

$$T_{\boldsymbol{\beta}}\mathbb{P}^N = \left\{\boldsymbol{u}_{\boldsymbol{\beta}} : \boldsymbol{u}_{\boldsymbol{\beta}}^T\mathbf{1} = 0\right\}. \tag{12}$$

A. Riemannian Gradient

Let the Riemannian gradient of a scalar function $F(\boldsymbol{\beta})$ on $\mathbb{P}^N$ be denoted by $\text{grad}F(\boldsymbol{\beta})$, and its classical gradient as seen in the Euclidean space as $\text{Grad}F(\boldsymbol{\beta})$. Its gradient in $\Re^N$ endowed with the metric g is scaled as $\text{Grad}F(\boldsymbol{\beta}) \odot \boldsymbol{\beta}$, where $\odot$ is element-wise multiplication. We then have [31]

$$\text{grad}F(\boldsymbol{\beta}) = \text{Proj}_{\boldsymbol{\beta}}(\text{Grad}F(\boldsymbol{\beta}) \odot \boldsymbol{\beta}) \tag{13}$$

where $\text{Proj}_{\boldsymbol{\beta}}(z)$ is the orthogonal projection into tangent space, which can be computed as

$$\text{Proj}_{\boldsymbol{\beta}}(z) = z - \alpha\boldsymbol{\beta} \tag{14}$$

where $\alpha = z^T\mathbf{1}$.

As a canonical way of identifying nearby tangent spaces, the Riemannian connection is typically given in the form of a covariant derivative which specifies how a tangent vector $\boldsymbol{v}_{\boldsymbol{\beta}}$ varies along the direction of another tangent vector $\boldsymbol{u}_{\boldsymbol{\beta}}$ on the manifold $\mathbb{P}^N$. To compute the Riemannian Hessian [24, Sec. 5.5], we need to use the notion of Riemannian connection on the manifold $\mathbb{P}^N$ denoted by $\nabla_{\boldsymbol{u}_{\boldsymbol{\beta}}}\boldsymbol{v}_{\boldsymbol{\beta}}$. Since $\mathbb{P}^N$ is a Riemannian submanifold of $\Re^N$ endowed with the metric g, the connection can be computed via

$$\nabla_{\boldsymbol{u}_{\boldsymbol{\beta}}}\boldsymbol{v}_{\boldsymbol{\beta}} = \text{Proj}_{\boldsymbol{\beta}}(\overline{\nabla}_{\boldsymbol{u}_{\boldsymbol{\beta}}}\boldsymbol{v}_{\boldsymbol{\beta}}) \tag{15}$$

where $\overline{\nabla}_{\boldsymbol{u}_{\boldsymbol{\beta}}}\boldsymbol{v}_{\boldsymbol{\beta}}$ is the connection on the manifold $\Re^N$ endowed with the metric g. The connection $\overline{\nabla}_{\boldsymbol{u}_{\boldsymbol{\beta}}}\boldsymbol{v}_{\boldsymbol{\beta}}$ in $\Re^N$ is computed using the Koszul formula, and after a few steps of computations it admits of matrix characterization [32]

$$\overline{\nabla}_{\boldsymbol{u}_{\boldsymbol{\beta}}}\boldsymbol{v}_{\boldsymbol{\beta}} = D\boldsymbol{v}_{\boldsymbol{\beta}}[\boldsymbol{u}_{\boldsymbol{\beta}}] - \frac{1}{2}(\boldsymbol{u}_{\boldsymbol{\beta}} \odot \boldsymbol{v}_{\boldsymbol{\beta}}) \oslash \boldsymbol{\beta} \tag{16}$$

where $\oslash$ denotes the elementwise division. $D\boldsymbol{v}_{\boldsymbol{\beta}}[\boldsymbol{u}_{\boldsymbol{\beta}}]$ is the Euclidian direction derivative of $\boldsymbol{v}_{\boldsymbol{\beta}}$ in the direction of $\boldsymbol{u}_{\boldsymbol{\beta}}$.

B. Riemannian Hessian

The Riemannian Hessian of $F(\boldsymbol{\beta})$ is defined as the connection

$$\begin{aligned}&\text{Hess}F(\boldsymbol{\beta})[\boldsymbol{u}_{\boldsymbol{\beta}}]\\ &\quad= \nabla_{\boldsymbol{u}_{\boldsymbol{\beta}}}\text{grad}F(\boldsymbol{\beta})\\ &\quad= \text{Proj}_{\boldsymbol{\beta}}\left(D\text{grad}F(\boldsymbol{\beta})[\boldsymbol{u}_{\boldsymbol{\beta}}] - \frac{1}{2}(\boldsymbol{u}_{\boldsymbol{\beta}} \odot \text{grad}F(\boldsymbol{\beta})) \oslash \boldsymbol{\beta}\right)\end{aligned} \tag{17}$$

where $D\text{grad}F(\boldsymbol{\beta})[\boldsymbol{u}_{\boldsymbol{\beta}}]$ is the Euclidean gradient directional derivative of the Riemannian gradient in the direction of $\boldsymbol{u}_{\boldsymbol{\beta}} \in T_{\boldsymbol{\beta}}\mathbb{P}^N$, which is calculated as

$$\begin{aligned}&D\text{grad}F(\boldsymbol{\beta})[\boldsymbol{u}_{\boldsymbol{\beta}}]\\ &\quad= D\text{Proj}_{\boldsymbol{\beta}}(\text{Grad}F(\boldsymbol{\beta}) \odot \boldsymbol{\beta})[\boldsymbol{u}_{\boldsymbol{\beta}}]\\ &\quad= D\text{Grad}F(\boldsymbol{\beta}) \odot \boldsymbol{\beta}[\boldsymbol{u}_{\boldsymbol{\beta}}] - D(\text{Grad}F(\boldsymbol{\beta}) \odot \boldsymbol{\beta})^T\mathbf{1}\boldsymbol{\beta}[\boldsymbol{u}_{\boldsymbol{\beta}}]\\ &\quad= D\text{Grad}F(\boldsymbol{\beta})[\boldsymbol{u}_{\boldsymbol{\beta}}] \odot \boldsymbol{\beta} + \text{Grad}F(\boldsymbol{\beta}) \odot \boldsymbol{u}_{\boldsymbol{\beta}}\\ &\qquad-(\text{Grad}F(\boldsymbol{\beta}) \odot \boldsymbol{\beta})^T\mathbf{1}\boldsymbol{u}_{\boldsymbol{\beta}} - D(\text{Grad}F(\boldsymbol{\beta}) \odot \boldsymbol{\beta})^T\mathbf{1}[\boldsymbol{u}_{\boldsymbol{\beta}}]\boldsymbol{\beta}\\ &\quad= D\text{Grad}F(\boldsymbol{\beta})[\boldsymbol{u}_{\boldsymbol{\beta}}] \odot \boldsymbol{\beta} + \text{Grad}F(\boldsymbol{\beta}) \odot \boldsymbol{u}_{\boldsymbol{\beta}}\\ &\qquad-(\text{Grad}F(\boldsymbol{\beta}) \odot \boldsymbol{\beta})^T\mathbf{1}\boldsymbol{u}_{\boldsymbol{\beta}} - (\text{Grad}F(\boldsymbol{\beta}) \odot \boldsymbol{u}_{\boldsymbol{\beta}})^T\mathbf{1}\boldsymbol{\beta}\\ &\qquad-(D\text{Grad}F(\boldsymbol{\beta})[\boldsymbol{u}_{\boldsymbol{\beta}}] \odot \boldsymbol{\beta})^T\mathbf{1}\boldsymbol{\beta}\end{aligned} \tag{18}$$

by making use of (14) and (15), where $D\text{Grad}F(\boldsymbol{\beta})[\boldsymbol{u}_{\boldsymbol{\beta}}]$ is the Euclidean directional derivative of the Euclidean gradient $\text{Grad}F(\boldsymbol{\beta})$ in the direction $\boldsymbol{u}_{\boldsymbol{\beta}} \in T_{\boldsymbol{\beta}}\mathbb{P}^N$.

C. Retraction Mapping

An important concept in the recent retraction-based framework of Riemannian optimization is retraction mapping [24, Sec. 4.1]. The exponential map $\text{Exp}_{\boldsymbol{\beta}}$ is the canonical choice for the retraction mapping; however, in this brief, we propose using the following standard approximation as the retraction mapping:

$$\boldsymbol{\beta}_t = R_{\boldsymbol{\beta}}(t\boldsymbol{u}_{\boldsymbol{\beta}}) := \frac{(\boldsymbol{\beta} \odot \exp(t(\boldsymbol{u}_{\boldsymbol{\beta}} \oslash \boldsymbol{\beta})))}{\left(\mathbf{1}^T(\boldsymbol{\beta} \odot \exp(t(\boldsymbol{u}_{\boldsymbol{\beta}} \oslash \boldsymbol{\beta})))\right)}$$

where t is called the step size, $\exp(\cdot)$ is an operator applied to matrices element by element. The retraction mapping is used to locate the next iterate on the manifold along a specified tangent vector, such as a search direction in line search in Newton's algorithm or the suboptimal tangent direction in the trust-region algorithm, see [24, Ch. 7].

IV. SPARSE DENSITY ESTIMATION USING RIEMANNIAN TRUST-REGION ALGORITHM

Since the constraint set of (2) is the multinomial manifold, our Riemannian optimization problem is simply formulated as

$$\min_{\boldsymbol{\beta}\in\mathbb{P}^N}\left\{F(\boldsymbol{\beta}) = \frac{1}{2}\boldsymbol{\beta}^T\boldsymbol{C}\boldsymbol{\beta} - \boldsymbol{\beta}^T\boldsymbol{p}\right\}. \tag{19}$$

The RTR algorithm retains the superlinearly convergent properties of the Euclidean trust-region method of a second-order algorithm, and it is suitable for large-scale optimization on Riemannian manifolds [33], [34]. Each iteration consists of two steps: 1) approximating the solution of the so-called trust-region subproblem and 2) computing a new iterate based on a retracting mapping. The trust region subproblem is given by

$$\begin{aligned}\min_{\boldsymbol{u}_{\boldsymbol{\beta}}\in T_{\boldsymbol{\beta}}\mathbb{P}^N, \|\boldsymbol{u}_{\boldsymbol{\beta}}\|\le\Delta}\ & F(\boldsymbol{\beta}) + g(\text{grad}F(\boldsymbol{\beta}), \boldsymbol{u}_{\boldsymbol{\beta}})\\ &+\frac{1}{2}g(\text{Hess}F(\boldsymbol{\beta})[\boldsymbol{u}_{\boldsymbol{\beta}}], \boldsymbol{u}_{\boldsymbol{\beta}})\end{aligned} \tag{20}$$

Require: δ, ρ, D_N, and the initial guess $\boldsymbol{\beta}_0 = [\frac{1}{N}, \cdots, \frac{1}{N}]^{\mathrm{T}}$ on the manifold $\mathbb{P}^N$;
Ensure: $\boldsymbol{\beta}$ that yields the minimum $F(\boldsymbol{\beta})$.
1: Calculate $\boldsymbol{C}$ and $\boldsymbol{p}$ based on δ, ρ, D_N;
2: Continue the following for loop until a convergence criterion is satisfied:
3: **for** $i = 1, 2, \ldots$ **do**
4: Approximately minimize the Trust-Region subproblem (20) for a new direction $\boldsymbol{u}_{\boldsymbol{\beta}}$;
5: Construct the new trial iterate by using retraction mapping $\boldsymbol{\beta}_t = R_{\boldsymbol{\beta}}(t\boldsymbol{u}_{\boldsymbol{\beta}})$
6: Update the iterate by rejecting or accepting $\boldsymbol{\beta}_t$ depending on its quality
7: Update the Trust-Region radius Δ.
8: **end for**

Fig. 1. Riemannian trust-region algorithm for sparse density estimation, referred to as RTR-MISE algorithm.

where Δ is an appropriate trust-region radius and $\|\boldsymbol{u}_{\boldsymbol{\beta}}\| = g(\boldsymbol{u}_{\boldsymbol{\beta}}, \boldsymbol{u}_{\boldsymbol{\beta}})$. For our objective function, it is easy to check that Euclidean gradient $\mathrm{Grad}F(\boldsymbol{\beta})$ and Euclidean Hessian $D\mathrm{Grad}F(\boldsymbol{\beta})[\boldsymbol{u}_{\boldsymbol{\beta}}]$ can be calculated as

$$\mathrm{Grad}F(\boldsymbol{\beta}) = \boldsymbol{C}\boldsymbol{\beta} - \boldsymbol{p} \tag{21}$$

and

$$D\mathrm{Grad}F(\boldsymbol{\beta})[\boldsymbol{u}_{\boldsymbol{\beta}}] = \boldsymbol{C}\boldsymbol{u}_{\boldsymbol{\beta}} \tag{22}$$

from which the Riemannian gradient and Hessian of the objective function $F(\boldsymbol{\beta})$ on the multinomial manifold can be calculated according to (13), (17), and (18).

With all the ingredients already, we can form the following algorithm for sparse density estimation in Fig. 1. The RTR algorithm is well implemented in the Manifold Optimization Toolbox Manopt `http://www.manopt.org` [35].

V. Simulation Study

Four numerical examples are provided in this section, with the first two as simulated probability density estimation experiments. The proposed approach is applied to solve the classification problem in the other two examples. In each of the first two examples, a data set of $N = 500$ points was randomly drawn from a known distribution $p(\boldsymbol{x})$ and used to construct the PDF $\hat{p}(\boldsymbol{x}; \boldsymbol{\beta}, \rho)$ using the proposed RTR-MISE approach. A separate test data set of $N_{\text{test}} = 10\,000$ points was used for evaluation according to

$$L_1 = \frac{1}{N_{\text{test}}} \sum_{k=1}^{N_{\text{test}}} |p(\boldsymbol{x}_k) - \hat{p}(\boldsymbol{x}_k; \boldsymbol{\beta}, \rho)|. \tag{23}$$

The experiment was repeated for 100 different random runs. The other five methods used for comparison for the first two examples are: 1) the well-known PW estimator; 2) the sparse density construction (SDC) algorithm [18]; 3) the sparse kernel density construction (SKD) algorithm [19]; 4) the RSDE with multiplicative nonnegative quadratic programming (RSDE-MNQP) [20]; and 5) the FCR-MISE algorithm [23]. We briefly point out the basic ideas in the above algorithms 2)–5). The SDC algorithm [18] is a regression-based pdf estimation method, in which the empirical cdf is constructed and used as the desired response. The sparse pdf is constructed by selecting one kernel forwardly. The SKD algorithm is also a regression-based pdf estimation method, in which the PW estimator is constructed and used as the desired response [19]. The RSDE-MNQP algorithm solves problem (7) using the MNQP algorithm [20]. Also aimed at solving (7), the FCR-MISE algorithm [23] formulates the density estimation in a forward constrained regression manner by selecting one kernel forwardly. We also point out that MISE cost function is used in PW estimator using grid search for an optimal kernel width. However, the kernel width for the other algorithm needs to be preset since its optimization process will become computational prohibitive.

TABLE II
Performance of Kernel Density Estimates for (a) Example 1 and (b) Example 2

(a)

Method	L_1 test error (mean ± STD)	Kernel numbers (mean ± STD)
PW	$(4.18 \pm 0.8) \times 10^{-3}$	500 ± 0
SDC [18]	$(3.83 \pm 0.8) \times 10^{-3}$	11.9 ± 2.6
SKD [19]	$(3.84 \pm 0.8) \times 10^{-3}$	15.3 ± 3.9
RSDE-MNQP	$(4.24 \pm 0.8) \times 10^{-3}$	129.4 ± 35.7
FCR-MISE [23]	$(3.33 \pm 0.8) \times 10^{-3}$	25.1 ± 2.7
RTR-MISE	$\mathbf{(3.13 \pm 0.7) \times 10^{-3}}$	36.7 ± 11.3

(b)

Method	L_1 test error (mean ± STD)	Kernel numbers (mean ± STD)
PW	$(3.18 \pm 0.13) \times 10^{-5}$	600 ± 0
SDC [18]	$(4.48 \pm 1.2) \times 10^{-5}$	14.9 ± 2.1
SKD [19]	$(3.11 \pm 0.5) \times 10^{-5}$	9.4 ± 1.9
RSDE-MNQP	$(3.67 \pm 0.7) \times 10^{-5}$	29.4 ± 10.1
FCR-MISE [23]	$(2.82 \pm 0.1) \times 10^{-5}$	19.4 ± 0.9
RTR-MISE	$\mathbf{(2.53 \pm 0.1) \times 10^{-5}}$	81.2 ± 20

Example 1: The density to be estimated for this 2-D example was given by the mixture of two densities of a Gaussian and a Laplacian, as defined by

$$p(\boldsymbol{x}) = \frac{1}{4\pi} \exp\left(-\frac{(x_1 - 2)^2}{2}\right) \exp\left(-\frac{(x_2 - 2)^2}{2}\right) + \frac{0.35}{8} \exp(-0.7|x_1 + 2|) \exp(-0.5|x_2 + 2|). \tag{24}$$

Example 2: The density to be estimated for this 6-D example was defined by

$$p(\boldsymbol{x}) = \frac{1}{3} \sum_{i=1}^{3} \frac{1}{(2\pi)^3 \sqrt{\det(\boldsymbol{\Gamma}_i)}} \times \exp\left(-\frac{1}{2}(\boldsymbol{x} - \boldsymbol{\mu}_i)^T \boldsymbol{\Gamma}_i^{-1} (\boldsymbol{x} - \boldsymbol{\mu}_i)\right) \tag{25}$$

with $\boldsymbol{\mu}_1 = [1.0, 1.0, 1.0, 1.0, 1.0, 1.0]^T$, $\boldsymbol{\mu}_2 = [-1.0, -1.0, -1.0, -1.0, -1.0, -1.0]^T$, $\boldsymbol{\mu}_3 = [0, 0, 0, 0, 0, 0]^T$, $\boldsymbol{\Gamma}_1 = \mathrm{diag}\{1.0, 2.0, 1.0, 2.0, 1.0, 2.0\}$ $\boldsymbol{\Gamma}_2 = \mathrm{diag}\{2.0, 1.0, 2.0, 1.0, 2.0, 1.0\}$, and $\boldsymbol{\Gamma}_3 = \mathrm{diag}\{2.0, 1.0, 2.0, 1.0, 2.0, 1.0\}$.

The results for Examples 1 and 2 were listed in Table II(a) and (b), respectively. For the PW pdf estimator, the kernel width was determined by MSIE [2]. For RSDE-MNQP, FCR-MISE, and the proposed RTR-MISE algorithm, the kernel width was empirically set. The results for SDC, SKD, and FCR-MISE are quoted from [18], [19], and [23], respectively. For the proposed RTR-MISE algorithm, we set $\delta = 10^{-5}$ for both examples. It is seen that the proposed algorithm can construct sparse kernel density estimators with an accuracy competitive with those of a PW estimator and other existing sparse kernel density estimators.

Provided a training data set of multiclass classification data sets C_j, $j = 1, \ldots, M$, respectively, the proposed method is readily applicable to the estimation of M conditional probability density

TABLE III
PERFORMANCE COMPARISON FOR EXAMPLE 3

Method	Kernel width	Test error rate
PW	0.24	8.1%
FCR-MISE [23]	0.13	8.3%
RTR-MISE	0.3	7.9%

functions $\hat{p}(\boldsymbol{x}; \boldsymbol{\beta}_{C_j}, \rho_{C_j}|C_j)$, respectively. The Bayes decision rule given by

$$\boldsymbol{x} \text{ belongs to } C_k, \quad k = \arg\max_j \hat{p}(\boldsymbol{x}; \boldsymbol{\beta}_{C_j}, \rho_{C_j}|C_j) \tag{26}$$

can be applied to the test data set to obtain the corresponding classification error rate as demonstrated in the following two examples.

Example 3: 2-D synthetic data set [36]. The data set was taken from http://www.stats.ox.ac.uk/PRNN/. The training set has 250 points with 125 points for each class (class 0: C_0 and class 1: C_1). The test set has 1000 points with 500 points for each class. The optimal Bayes error rate based on the true probability distribution is known to be 8%. For the same data set, the test error rate of 10.6% and 9.3% have been reported for a support vector machine using 38 Gaussian kernels and a relevance vector machine, using four Gaussian kernels, respectively [37]. Table III lists the classification results obtained by the three kernel density estimators, the PW and FCR-MISE, and the proposed RTR-MISE algorithm. The widths for these algorithms were set empirically by minimizing the test error rate. We point out that the proposed RTR-MISE algorithm selects only two or three kernels for conditional pdf estimator for each class, while the PW-based conditional pdf of each class has 125 kernels using the full training data set. Clearly, the proposed RTR-MSIE algorithm has a classification performance comparable with those of PW, with all being very close to the known optimal Bayes error rate.

Example 4: Optical recognition of handwritten digits [38]. The data set was created by extracting normalized bitmaps of handwritten digits from a preprinted form. From a total of 43 people, 30 contributed to the training set and different 13 to the test set. 32×32 bitmaps are divided into nonoverlapping blocks of 4×4 and the number of on pixels are counted in each block. This generates an input matrix of 8×8, resulting in 64 input features. The produced data set has 10 classes, and the training/test set has 3823 and 1797 data points, both with balanced class distribution. The accuracy on the testing sets [38] with k-nn using Euclidean distance as the metric ranges from 97.38% for $k = 2\%$ to 98% for $k = 1$. We extract training data set for each class and applied the proposed algorithm with/without initialization, respectively, as $\hat{p}(\boldsymbol{x}; \boldsymbol{\beta}_{C_j}, \rho_{C_j}|C_j)$, with $\rho_{C_j} = 5$. Equation (26) is used to obtain the predicted class labels. In each case (with/without) initialization the classification accuracy achieved 98%, which is comparable with that of the nearest neighbor approach ($k = 1$). For this example, when equal weighting initialization is applied, the resultant models are actually obtained as nonsparse PW estimators. However, in the case of no initialization, i.e., when the default option is set in the RTR algorithm, 10 sparser models are obtained using 258/376, 263/389, 260/380, 263/389, 262/387, 258/376, 258/377, 262/387, 260/380, and 258/382 of the data points.

VI. CONCLUSION

We have introduced a new sparse kernel density estimator for the finite mixture model based on the MISE criterion. The recently established RTR algorithm is used by exploiting the multinomial manifold, which forms the parameter search space of the mixing coefficients of the finite mixture model. We have derived the first- and second-order Riemannian gradients on multinomial manifold that are needed to implement the RTR algorithm. Numerical examples are employed to demonstrate the effectiveness of the proposed approach with an accuracy competitive with those of existing kernel density estimators.

REFERENCES

[1] G. McLachlan and D. Peel, *Finite Mixture Models*. New York, NY, USA: Wiley, 2000.

[2] B. W. Silverman, *Density Estimation for Statistics and Data Analysis*. London, U.K.: Chapman & Hall, 1986.

[3] L. Rutkowski, "Adaptive probabilistic neural networks for pattern classification in time-varying environment," *IEEE Trans. Neural Netw.*, vol. 15, no. 4, pp. 811–827, Jul. 2004.

[4] H. Yin and N. M. Allinson, "Self-organizing mixture networks for probability density estimation," *IEEE Trans. Neural Netw.*, vol. 12, no. 2, pp. 405–411, Mar. 2001.

[5] Z. Halbe, M. Bortman, and M. Aladjem, "Regularized mixture density estimation with an analytical setting of shrinkage intensities," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 24, no. 3, pp. 460–473, Mar. 2013.

[6] K. Zhang and J. T. Kwok, "Simplifying mixture models through function approximation," *IEEE Trans. Neural Netw.*, vol. 21, no. 4, pp. 644–658, Apr. 2010.

[7] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *J. Roy. Statist. Soc. B (Methodological)*, vol. 39, no. 1, pp. 1–38, 1977.

[8] J. A. Bilmes, "A gentle tutorial of the EM algorithm and its application to parameter estimation for Gaussian mixture and hidden Markov models," Dept. Elect. Eng. Comput. Sci., Univ. California, Berkeley, Berkeley, CA, USA, Tech. Rep. ICSI-TR-97-021, 1998.

[9] B. Efron and R. J. Tibshirani, *An Introduction to the Bootstrap*. London, U.K.: Chapman & Hall, 1993.

[10] Z. R. Yang and S. Chen, "Robust maximum likelihood training of heteroscedastic probabilistic neural networks," *Neural Netw.*, vol. 11, no. 4, pp. 739–747, Jun. 1998.

[11] E. Parzen, "On estimation of a probability density function and mode," *Ann. Math. Statist.*, vol. 33, no. 3, pp. 1066–1076, Sep. 1962.

[12] J. Weston, A. Gammerman, M. O. Stitson, V. Vapnik, V. Vovk, and C. Watkins, "Support vector density estimation," in *Advances in Kernel Methods: Support Vector Learning*, B. Schölkopf, C. J. C. Burges, and A. J. Smola, Eds. Cambridge, MA, USA: MIT Pres, 1999, pp. 293–306.

[13] V. N. Vapnik and S. Mukherjee, "Support vector method for multivariate density estimation," in *Advances in Neural Information Processing Systems*, S. A. Solla, T. K. Leen, and K. Müller, Eds. Cambridge, MA, USA: MIT Press, 2000, pp. 659–665.

[14] A. Choudhury, "Fast machine learning algorithms for large data," Ph.D. dissertation, School Eng. Sci., Univ. Southampton, Southampton, U.K., 2002.

[15] S. Chen, S. A. Billings, and W. Luo, "Orthogonal least squares methods and their application to non-linear system identification," *Int. J. Control*, vol. 50, no. 5, pp. 1873–1896, 1989.

[16] X. Hong, P. M. Sharkey, and K. Warwick, "Automatic nonlinear predictive model-construction algorithm using forward regression and the PRESS statistic," *IEE Proc.-Control Theory Appl.*, vol. 150, no. 3, pp. 245–254, May 2003.

[17] S. Chen, X. Hong, C. J. Harris, and P. M. Sharkey, "Sparse modeling using orthogonal forward regression with PRESS statistic and regularization," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 34, no. 2, pp. 898–911, Apr. 2004.

[18] S. Chen, X. Hong, and C. J. Harris, "Sparse kernel density construction using orthogonal forward regression with leave-one-out test score and local regularization," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 34, no. 4, pp. 1708–1717, Aug. 2004.

[19] S. Chen, X. Hong, and C. J. Harris, "An orthogonal forward regression technique for sparse kernel density estimation," *Neurocomputing*, vol. 71, nos. 4–6, pp. 931–943, Jan. 2008.

[20] M. Girolami and C. He, "Probability density estimation from optimally condensed data samples," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 10, pp. 1253–1264, Oct. 2003.

[21] D. W. Scott, "Parametric statistical modeling by minimum integrated square error," *Technometrics*, vol. 43, no. 3, pp. 274–285, Aug. 2001.

[22] X. Hong and C. J. Harris, "A mixture of experts network structure construction algorithm for modelling and control," *Appl. Intell.*, vol. 16, no. 1, pp. 59–69, 2002.

[23] X. Hong, S. Chen, A. Qatawneh, K. Daqrouq, M. Sheikh, and A. Morfeq, "Sparse probability density function estimation using the minimum integrated square error," *Neurocomputing*, vol. 115, pp. 122–129, Sep. 2013.

[24] P.-A. Absil, R. Mahony, and R. Sepulchre, *Optimization Algorithms on Matrix Manifolds*. Princeton, NJ, USA: Princeton Univ. Press, 2008.

[25] B. Mishra, G. Meyer, F. Bach, and R. Sepulchre, "Low-rank optimization with trace norm penalty," *SIAM J. Optim.*, vol. 23, no. 4, pp. 2124–2149, 2013.

[26] M. Harandi, R. Hartley, C. Shen, B. Lovell, and C. Sanderson. (2014). "Extrinsic methods for coding and dictionary learning on Grassmann manifolds." [Online]. Available: http://arxiv.org/abs/1401.8126

[27] Y. M. Lui, "Advances in matrix manifolds for computer vision," *Image Vis. Comput.*, vol. 30, nos. 6–7, pp. 380–388, 2012.

[28] J. Weston, A. Ellisseeff, B. Schölkopf, and M. Tipping, "Use of the zero norm with linear models and kernel methods," *J. Mach. Learn. Res.*, vol. 3, no. 3, pp. 1439–1461, 2003.

[29] X. Hong, S. Chen, and C. J. Harris, "Using zero-norm constraint for sparse probability density function estimation," *Int. J. Syst. Sci.*, vol. 43, no. 11, pp. 2107–2113, 2012.

[30] R. Inokuchi and S. Miyamoto, "*c*-means clustering on the multinomial manifold," in *Modeling Decisions for Artificial Intelligence* (Lecture Notes in Computer Science), vol. 4617. Berlin, Germany: Springer-Verlag, 2007, pp. 261–268.

[31] Y. Sun, J. Gao, X. Hong, B. Mishra, and B. Yin, "Heterogeneous tensor decomposition for clustering via manifold optimization," to be published.

[32] B. Vandereycken, "Riemannian and multilevel optimization for rank-constrained matrix problems," Ph.D. dissertation, Faculty Eng., Katholieke Univ. Leuven, Leuven, Belgium, 2010.

[33] C. G. Baker, "Riemannian manifold trust-region methods with applications to eigenproblems," Ph.D. dissertation, School Comput. Sci., Florida State Univ., Tallahassee, FL, USA, 2008.

[34] B. Misha and R. Sepulchre. (2014). "Riemannian preconditioning." [Online]. Available: http://arxiv.org/abs/1405.6055

[35] N. Boumal, B. Mishra, P.-A. Absil, and R. Sepulchre. (2013). "Manopt, a MATLAB toolbox for optimization on manifolds." [Online]. Available: http://arxiv.org/abs/1308.5200

[36] B. D. Ripley, *Pattern Recognition and Neural Networks*. Cambridge, MA, USA: Cambridge Univ. Press, 1996.

[37] M. E. Tipping, "Sparse Bayesian learning and the relevance vector machine," *J. Mach. Learn. Res.*, vol. 1, pp. 211–244, Jun. 2001.

[38] K. Bache and M. Lichman. (2013). "UCI machine learning repository," School Inf. Comput. Sci., Univ. California, Irvine, Irvine, CA, USA. [Online]. Available: http://archive.ics.uci.edu/ml