

DEPARTMENT OF STATISTICS

University of Wisconsin

1300 University Avenue

Madison, WI 53706

TECHNICAL REPORT NO. 1100

December 28, 2004

An Effective Method for High Dimensional Log-density ANOVA  
Estimation, with Application to Nonparametric Graphical Model Building

Yongho Jeon & Yi Lin

Department of Statistics, University of Wisconsin, Madison WI

This research was supported in part by National Science Foundation grant DMS 0134987.

# An Effective Method for High Dimensional Log-density ANOVA Estimation, with Application to Nonparametric Graphical Model Building

BY YONGHO JEON AND YI LIN

*Department of Statistics, University of Wisconsin–Madison*

yjeon@stat.wisc.edu yilin@stat.wisc.edu

## Abstract

The log-density functional ANOVA model provides a powerful framework for the estimation and interpretation of high dimensional densities. Existing methods for fitting such a model require repeated numerical integration of high dimensional functions, and are infeasible in problems of dimension larger than four. We propose a new method for fitting the log-density ANOVA model based on a penalized  $M$ -estimation formulation with a novel loss function. Solving the penalized  $M$ -estimation problem does not require high dimensional integration: only one dimensional integrals are required and they can be computed quickly by using the cumulative distribution function of familiar one dimensional densities. Simulations indicate that the proposed method is statistically very efficient and computationally feasible in high dimensional problems. We apply the new method to the construction and estimation of (undirected) nonparametric graphical models. The graphical models use graphs to display the conditional dependence among random variables and have become very popular, but have mostly been studied parametrically. Our method provides a practical way to construct and estimate nonparametric graphical models and handles the continuous and categorical variables in a unified fashion.

## 1 Introduction

Consider the density estimation problem in which we are given a random sample of a  $d$ -dimensional random vector  $X = (X_1, \dots, X_d)$ , and we wish to estimate the density function  $p(\cdot)$  of  $X$ . A number of nonparametric algorithms are successful for low dimensional problems ( $d \leq 3$ ), but there are few practical algorithms for higher dimensional problems. A major difficulty is that a general high dimensional density function is hard to estimate,

both in terms of accuracy and computational cost. Even when an accurate estimate is available, a complicated high dimensional density function can be very hard to interpret.

The log-density smoothing spline ANOVA model provides a powerful framework for the estimation and interpretation of high dimensional density functions. In such a model the log-density function is decomposed as a sum of a constant term, one dimensional functions (main effects), two dimensional functions (two-way interactions), and so on:

$$\eta(\mathbf{x}) = \text{constant} + \sum_{j=1}^d \eta_j(x_j) + \sum_{j < k} \eta_{jk}(x_j, x_k) + \dots, \quad (1.1)$$

where the components satisfy side conditions that guarantee uniqueness, and the series is usually truncated in some manner to enhance interpretability. For an overview of such models, see Gu (2002). Notice that the additive log-density model (with no interaction terms) actually assumes independence among the variables. The all-two-way-interaction model (the model with all the main effects and two-way interactions, but no higher order interactions) is the simplest model in which dependence structure can be incorporated, and is commonly used. Such a model has the further advantage that the components in the ANOVA decomposition can be visualized. In this paper we will mainly consider the all-two-way-interaction model, though the new method that we are to propose for fitting log-density ANOVA model is applicable to more general log-density ANOVA structures.

There is a close connection between the log-density ANOVA model and the (undirected) graphical models, which use graphs to intuitively represent the conditional dependence structure among a number of variables. For example, in a three dimensional problem, the absence of the terms  $\eta_{23}$  and  $\eta_{123}$  in the log-density ANOVA decomposition (1.1) indicates that the random variables  $X_2$  and  $X_3$  are conditionally independent given  $X_1$ . The aforementioned three dimensional example can be represented by the graph

$$X_2 \text{---} X_1 \text{---} X_3.$$

Currently most of the research on undirected graphical models has been parametric. When the variables considered are categorical, graphical models are special cases of the loglinear models; when the variables are continuous, the current research on graphical mod-

els assumes joint Gaussian distribution for the variables; when there are both categorical and continuous variables, a conditional Gaussian distribution is usually assumed. Model selection among graphical models are typically done with stepwise forward/backward type procedures. For a review of graphical models, see, for example, Edwards (2000). To enhance the scope of applicability of the graphical model methodology, in this paper we will consider the building of (undirected) nonparametric graphical models through their connection with log-density ANOVA models. The relation between these two types of models is particularly simple when we concentrate on the log-density all-two-way-interaction model. In such a model, the absence of interaction term between any two variables entails conditional independence between the two variables given all other variables, and there is a one-to-one correspondence between the submodels of the log-density all-two-way-interaction model with graphical displays. The log-density all-two-way-interaction model can be seen as the continuous counterpart to the the all-two-way-interaction loglinear model discussed in Whittaker (1990). Notice that the commonly used parametric Gaussian graphical model can also be seen as a parametric special case of the log-density all-two-way-interaction model.

The building of nonparametric graphical models via log-density ANOVA model depends crucially on the availability of effective algorithms to fit the log-density ANOVA model in high dimensions. Currently the most commonly used method for fitting the log-density ANOVA model is the penalized likelihood method. Leonard (1978) introduced the logistic density transform  $p = \exp(h) / \int \exp(h)$  to incorporate the positivity and unity constraints of density function, and proposed to estimate  $h$  via penalized log likelihood. Silverman (1982) proposed and studied the theoretical properties of the penalized likelihood estimator obtained by solving

$$\arg \min_{\eta} \left\{ -\frac{1}{n} \sum_{i=1}^n \eta(\mathbf{x}_i) + \int e^{\eta} + \lambda J(\eta) \right\} \quad (1.2)$$

over a reproducing kernel Hilbert space (RKHS)  $\mathcal{H}_R$ , where  $J$  is a penalty functional that involves only derivatives, usually a squared semi-norm in  $\mathcal{H}_R$ . Gu and Qiu (1993) studied the theoretical property of the penalized likelihood estimate of the logistic density over

reproducing kernel Hilbert spaces. This estimate is the solution to

$$\arg \min_{\eta} \left\{ -\frac{1}{n} \sum_{i=1}^n \eta(\mathbf{x}_i) + \log \int e^{\eta} + \lambda J(\eta) \right\}. \quad (1.3)$$

Gu (1993) provided a practical algorithm for (1.3), and gave some one and two dimensional examples. Gu (2002) and Gu and Wang (2003) improved Gu's original algorithm, and the new algorithm is available in the software R.

While the penalized likelihood method has been successful in low dimensional log-density ANOVA problems, it is not practically feasible for high dimensional problems. In high dimensional problems, the major difficulty of the penalized likelihood method is the calculation of  $\int \exp(\eta)$  involved in the estimation. Notice that in general  $\int \exp(\eta)$  does not decompose even when  $\eta$  is in an all-two-way interaction model. For each fixed tuning parameter, solving (1.2) or (1.3) involves a number of Newton-Raphson iterations, and an expensive high dimensional integration is needed for each of the iterations. In one or two dimensions, this integral may be approximated by using simple grid point cubature. However, the use of grid points is not practical in high dimension, as the number of points for a decent approximation increases exponentially with the dimension  $d$ , and the requirement on accuracy is high for the approximation to be used successfully in the Newton-Raphson iterations. A sparse grid method has been used in Gu and Wang (2003) for the integration. However, it still can not provide sufficient accuracy with reasonable number of points in high dimensional problems. Furthermore, the sparse grid cubature involves negative weights, which causes serious numerical problems since the integration is embedded in an optimization procedure. For instance, the minimand that is convex in theory is not guaranteed to be convex numerically. So far the highest dimensional log-density ANOVA problem tackled in the literature is of dimension four, and it usually takes a large amount of time for the penalized likelihood method to fit four dimensional problems.

In this paper we propose a new method that is suitable for log-density ANOVA model estimation in high dimensional space. This is a penalized  $M$ -estimation type method with a novel loss function that targets the true log-density. For each fixed tuning parameter, solving our penalized  $M$ -estimation formulation involves only one-dimensional integrals, and these

one-dimensional integrals can be computed quickly by using the cumulative distribution function of familiar one dimensional densities. The computational load of the new method is much lighter than that of the penalized likelihood method, and is practically feasible for high dimensional problems. The penalized  $M$ -estimation formulation of the proposed algorithm is given in Section 2. In Section 3 the function space for the log-density smoothing spline ANOVA model is briefly reviewed, and we also introduce an alternative penalty to the smoothing spline penalty. This is a sparsity-inducing penalty and when it is used in our formulation, the estimation and model selection in the log-density ANOVA model can be done simultaneously. This enables us to build and fit nonparametric graphical models. Section 4 gives the detailed algorithm of the penalized  $M$ -estimation method with the sparsity-inducing penalty in the log-density ANOVA model. Simulations and real examples are given in Section 5 and Section 6. We give some discussions in Section 7.

## 2 The penalized $M$ -estimation formulation of the new method

Let  $\mathcal{X}$  be the support of the density  $p(x)$  of  $X$  and  $\rho$  be a fixed positive density function over  $\mathcal{X}$ . We find  $f \in \mathcal{H}$  to minimize

$$l_n(f) + \lambda J(f), \tag{2.1}$$

where

$$l_n(f) = \frac{1}{n} \sum_{i=1}^n e^{-f(\mathbf{x}_i)} + \int f \rho.$$

Here  $\mathcal{H}$  is the reproducing kernel Hilbert space (typically Sobolev Hilbert spaces or tensor product spaces of them. More details on this function space are given in Section 3), and  $J$  is a penalty functional, usually a squared semi-norm in  $\mathcal{H}$ . The formulation (2.1) is of the form of the method of regularization. Cox and O'Sullivan (1990) provided a general framework for studying the theoretical properties of this type of method. In general, under mild conditions, the estimator from (2.1) converges to the minimizer of the population version of  $l_n$ , when the tuning parameter  $\lambda$  is chosen to go to zero with certain rates. The

population version of the  $l_n$  in our case is

$$l(f) = E\{e^{-f(\mathbf{X})}\} + \int f\rho = \int e^{-f}p + \int f\rho. \quad (2.2)$$

The first and second order Fréchet derivatives of  $l(f)$  are

$$\begin{aligned} Dl(f)h &= -\int e^{-f}hp + \int h\rho = \int h(\rho - e^{-f}p), \\ D^2l(f)gh &= \int e^{-f}ghp, \end{aligned}$$

where  $D$  denotes Fréchet derivative operator. By setting  $Dl(f)h$  to be zero for all  $h$ , we get  $\rho - \exp(-f)p = 0$ , that is,  $\exp(f)\rho = p$ . Also,  $l$  is strictly convex since  $D^2l(f)gg = \int e^{-f}g^2p > 0$  for any  $g \neq 0$  in the space  $\mathcal{H}$ . Therefore  $l(f)$  is uniquely minimized by  $f = \log p - \log \rho$ , and this is what our method estimates. A detailed study of the consistency properties of our method including the rates of convergence will be given in a separate paper. After obtaining the solution  $\hat{f}$  to the minimizing problem (2.1), the estimate of the density  $p$  is  $\exp(\hat{f})\rho$ .

The function  $\rho(\mathbf{x})$  is an *baseline density* that is chosen before the estimation, and  $\exp(\hat{f}(\mathbf{x}))$  is used to catch the detailed density. The baseline density can be chosen to be any density function with the same support as the density  $p(x)$ . The optimization problem (2.1) involves an integral  $\int f\rho$ . As we will see in the Section 4.1, with suitable choices of  $\rho$ , this integral naturally decomposes into a sum of integrals in low dimensions in the log-density ANOVA model, and these low dimensional integrals can be further decomposed into products of one dimensional integrals due to the properties of the reproducing kernels used in the log-density smoothing spline ANOVA model. The one dimensional integrals can be computed quickly by using cumulative distribution functions of familiar one dimensional distributions. Therefore the integration  $\int f\rho$  involved in the optimization procedure can be computed quickly and accurately.

Suppose that the function space  $\mathcal{H}$  can be decomposed into  $\mathcal{H} = \{1\} \oplus \mathcal{G}$ , where  $\{1\}$  is the constant space, and  $\mathcal{G}$  is its orthogonal complement. Then the minimization problem

(2.1) over  $f \in \mathcal{H}$  is equivalent to find  $g \in \mathcal{G}$  which minimizes

$$\log \left\{ \frac{1}{n} \sum_{i=1}^n e^{-g(\mathbf{x}_i)} \right\} + \int g\rho + \lambda J(g), \quad (2.3)$$

provided that the penalty  $J$  remains unchanged for adding a constant (which is typically true since  $J$  typically involves only derivatives), since

$$\begin{aligned} & \min_{f \in \mathcal{H}} \left\{ \frac{1}{n} \sum_{i=1}^n e^{-f(\mathbf{x}_i)} + \int f\rho + \lambda J(f) \right\} \\ &= \min_{g \in \mathcal{G}, d \in \mathbb{R}} \left\{ \frac{1}{n} \sum_{i=1}^n e^{-g(\mathbf{x}_i)} e^{-d} + \int g\rho + d + \lambda J(g) \right\} \\ &= \min_{g \in \mathcal{G}} \left\{ 1 + \int g\rho + \log \left( \frac{1}{n} \sum_{i=1}^n e^{-g(\mathbf{x}_i)} \right) + \lambda J(g) \right\}. \end{aligned}$$

If  $\hat{g}$  minimizes (2.3), then the estimator for the density is of the form

$$\hat{p}(\mathbf{x}) = \text{constant} \cdot e^{\hat{g}(\mathbf{x})} \cdot \rho(\mathbf{x}). \quad (2.4)$$

Here the constant term can be chosen to satisfy the unity constraint  $\int \hat{p}(\mathbf{x}) d\mathbf{x} = 1$ . After  $\hat{g}$  is obtained by solving (2.3), when the normalizing constant in (2.4) is desired, we need to compute a high dimensional integral. However, this step is separate from the optimization procedure, and is only needed once in the normalization after the estimation of  $\hat{g}$ . Therefore it does not cause serious computational problems.

### 3 Function space of smoothing spline ANOVA

Let  $\mathcal{H}^{(j)}$  be an RKHS of univariate functions on  $\mathcal{X}_j$  of the form  $\mathcal{H}^{(j)} = \{1^{(j)}\} \oplus \mathcal{H}_1^{(j)}$ , where  $\{1^{(j)}\}$  is the space of constant functions on  $\mathcal{X}_j$  and  $\mathcal{H}_1^{(j)}$  is its orthogonal complement. We can construct an RKHS of functions on  $\mathcal{X} = \mathcal{X}_1 \times \cdots \times \mathcal{X}_d$  through the tensor product space strategy:

$$\bigotimes_{j=1}^d \mathcal{H}^{(j)} = \bigotimes_{j=1}^d \{ \{1^{(j)}\} \oplus \mathcal{H}_1^{(j)} \} = \{1\} \oplus \left\{ \bigoplus_{j=1}^d \mathcal{H}_1^{(j)} \right\} \oplus \left\{ \bigoplus_{j < k} [\mathcal{H}_1^{(j)} \otimes \mathcal{H}_1^{(k)}] \right\} \oplus \cdots, \quad (3.1)$$

where  $\{1\}$  denotes the constant functions on  $\mathcal{X}$  and factors of the form  $\{1^{(j)}\}$  are omitted whenever they multiply a term of a different form with some abuse of notation.

When  $X_j$  is a continuous variable on the domain  $[0, 1]$ , we will take  $\mathcal{H}^{(j)}$  as the commonly used second order Sobolev-Hilbert space  $\{f|f, f' \text{ are absolutely continuous; } f'' \in L_2([0, 1])\}$ . If endowed with the inner product

$$\langle f_1, f_2 \rangle = \left\{ \int_0^1 f_1(t) dt \right\} \left\{ \int_0^1 f_2(t) dt \right\} + \left\{ \int_0^1 f_1'(t) dt \right\} \left\{ \int_0^1 f_2'(t) dt \right\} + \int_0^1 f_1''(t) f_2''(t) dt,$$

$\mathcal{H}^{(j)}$  is a RKHS with the RK  $R(s, t) = 1 + k_1(s)k_1(t) + k_2(s)k_2(t) - k_4(|s - t|)$ , where

$$\begin{aligned} k_1(x) &= x - \frac{1}{2} \\ k_2(x) &= \frac{1}{2} \left\{ k_1^2(x) - \frac{1}{12} \right\} \\ k_4(x) &= \frac{1}{24} \left\{ k_1^4(x) - \frac{1}{2} k_1^2(x) + \frac{7}{240} \right\}. \end{aligned}$$

$\mathcal{H}^{(j)}$  can be decomposed as  $\mathcal{H}^{(j)} = \{1^{(j)}\} \oplus \mathcal{H}_1^{(j)}$ . When  $X_j$  is a categorical variable on the discrete domain  $\{1, \dots, L\}$ , we have an orthogonal decomposition  $\mathcal{H}^{(j)} = \{1^{(j)}\} \oplus \mathcal{H}_1^{(j)}$  under the squared norm  $\|f\|^2 = L^{-1} \sum_{l=1}^L f^2(l)$ , where  $\mathcal{H}_1^{(j)}$  is a RKHS with the RK  $R_1(s, t) = LI(s = t) - 1$ . See Wahba (1990) and Gu (2002) for details.

In the log-density smoothing spline ANOVA model, the log-density is assumed to have an ANOVA decomposition with only low order interactions. If we choose the baseline density  $\rho$  such that  $\log \rho$  has an additive structure, as we will do in our implementation, then  $f = \log p - \log \rho$  has the same ANOVA structure as the log-density. In the smoothing spline ANOVA model, we assume each functional component in the decomposition (1.1) of  $f$  lies in a corresponding subspace in the orthogonal decomposition (3.1) of  $\bigotimes_{j=1}^d \mathcal{H}^{(j)}$ . Thus the function space  $\mathcal{H}$  assumed for  $f$  consists of some orthogonal component subspaces in (3.1). Relabeling the subspaces other than the null space  $\mathcal{N} = \{1\}$  in the model as  $\mathcal{G}^{(\alpha)}$ ,  $\alpha = 1, \dots, p$ , then  $\mathcal{H} = \mathcal{N} \oplus \left\{ \bigoplus_{\alpha=1}^p \mathcal{G}^{(\alpha)} \right\}$ , and the smoothing spline method finds  $f \in \mathcal{H}$  to minimize

$$l_n(f) + \lambda \sum_{\alpha=1}^p \theta_{\alpha}^{-1} \|P^{\alpha} f\|^2, \quad (3.2)$$

where  $P^\alpha$  is the orthogonal projector in  $\mathcal{H}$  into  $\mathcal{G}^{(\alpha)}$ . Here  $\theta_\alpha \geq 0$  and if  $\theta_\alpha = 0$  then the minimizer is taken to satisfy  $\|P^\alpha f\|^2 = 0$ . (We use the convention  $0/0 = 0$ .) It is known that the minimizer of (3.2) in the regression problem is in the finite dimensional space  $\mathcal{N} \oplus \mathcal{G}_n$ , where  $\mathcal{G}_n = \text{span}\{R_\theta(\mathbf{x}_i, \cdot), i = 1, \dots, n\}$  and  $R_\theta(s, t) = \sum_{\alpha=1}^p \theta_\alpha R_\alpha(s, t)$  (Wahba, 1990, Chapter 10).

In our log-density ANOVA model fitting, the smoothing spline method would find  $f \in \{1\} \oplus \mathcal{G}_n$  to minimize our new formulation (2.1) with  $J(f) = \sum_{\alpha=1}^p \theta_\alpha^{-1} \|P^\alpha f\|^2$  and it is equivalent to find  $g \in \mathcal{G}_n$  to minimize

$$\log \left\{ \frac{1}{n} \sum_{i=1}^n e^{-g(\mathbf{x}_i)} \right\} + \int g \rho + \lambda \sum_{\alpha=1}^p \theta_\alpha^{-1} \|P^\alpha g\|^2. \quad (3.3)$$

The COSSO (Lin and Zhang, 2002) is a method of regularization with the penalty functional being the sum of component norms, instead of the squared norms employed in the traditional smoothing spline method. The penalty used in COSSO enables us to get a sparse solution in terms of SS-ANOVA functional components so that both estimation and model selection can be carried out simultaneously.

The COSSO finds  $g \in \mathcal{G}$  to minimize

$$\log \left\{ \frac{1}{n} \sum_{i=1}^n e^{-g(\mathbf{x}_i)} \right\} + \int g \rho + \tau \sum_{\alpha=1}^p \|P^\alpha g\|. \quad (3.4)$$

It is easy to show with arguments similar to those in Lin and Zhang (2002) that the minimization problem (3.4) is equivalent to the problem of finding  $\theta = (\theta_1, \dots, \theta_p)^T$  and  $g \in \mathcal{G}$  to minimize

$$\log \left\{ \frac{1}{n} \sum_{i=1}^n e^{-g(\mathbf{x}_i)} \right\} + \int g \rho + \lambda_0 \sum_{\alpha=1}^p \theta_\alpha^{-1} \|P^\alpha g\|^2 + \lambda \sum_{\alpha=1}^p \theta_\alpha, \quad (3.5)$$

subject to  $\theta_\alpha \geq 0$ ,  $\alpha = 1, \dots, p$ , where  $\lambda_0$  is a fixed constant and  $\lambda$  is a smoothing parameter. Note that there is only one smoothing parameter  $\lambda$  in (3.5). The  $\theta_\alpha$  are not free smoothing parameters but part of the estimate.

For any fixed  $\theta$ , the COSSO (3.5) is a smoothing spline (3.3) problem with fixed smoothing parameters. Thus we find a solution to (3.5) among the functions of the form

$$g(\mathbf{x}) = \sum_{i=1}^n c_i R_{\theta}(\mathbf{x}_i, \mathbf{x}) = \sum_{\alpha=1}^p \theta_{\alpha} \sum_{i=1}^n c_i R_{\alpha}(\mathbf{x}_i, \mathbf{x}).$$

## 4 Algorithm

We consider the all-two-way-interaction log-density ANOVA model in our implementation. For the solution  $\hat{g}$  in the all-two-way-interaction model, the estimated log-density  $\hat{p}(\mathbf{x}) = \text{constant} + \hat{g}(\mathbf{x}) + \log \rho(\mathbf{x})$  holds two-way interaction structure if  $\log \rho$  is additive.

### 4.1 Baseline density function

We assume the domain of any continuous variable is  $[0, 1]$  and we use the Beta family for the baseline density. For each continuous variable  $X_j$ , we fit a Beta density to the marginal distribution of  $X_j$  with the method of maximum likelihood. Let  $\rho^{(j)}$  be the estimated density function. If categorical, let  $\rho^{(j)}$  be the empirical relative frequency. Then the product of marginal baseline densities  $\rho^{(j)}$  serves as the baseline density:

$$\rho(\mathbf{x}) = \prod_{j=1}^d \rho^{(j)}(x_j).$$

Let us now consider the all-two-way-interaction log-density ANOVA model. Denoting  $R_{jk}((x_j, x_k), (x'_j, x'_k)) = R_j(x_j, x'_j)R_k(x_k, x'_k)$  for  $1 \leq j < k \leq d$ , the solution  $g = \sum_{j=1}^d g_j + \sum_{j < k} g_{jk}$  to the problem of minimizing (3.3) is to be found among the functions of the form

$$g(\mathbf{x}) = \sum_{i=1}^n c_i \left\{ \sum_{j=1}^d \theta_j R_j(\mathbf{x}_i, \mathbf{x}) + \sum_{j < k} \theta_{jk} R_{jk}(\mathbf{x}_i, \mathbf{x}) \right\}.$$

The integral term can be computed by

$$\begin{aligned} \int g(\mathbf{x}) \rho(\mathbf{x}) d\mathbf{x} &= \sum_{i=1}^n c_i \left\{ \sum_{j=1}^d \theta_j \int R_j(\mathbf{x}_i, \mathbf{x}) \rho(\mathbf{x}) d\mathbf{x} + \sum_{j < k} \theta_{jk} \int R_{jk}(\mathbf{x}_i, \mathbf{x}) \rho(\mathbf{x}) d\mathbf{x} \right\} \\ &= \sum_{i=1}^n c_i \left\{ \sum_{j=1}^d \theta_j \tilde{b}_{ij} + \sum_{j < k} \theta_{jk} \tilde{b}_{ij} \tilde{b}_{ik} \right\}, \end{aligned}$$

where  $\tilde{b}_{ij} = \int R_j(\mathbf{x}_i, \mathbf{x}) \rho(\mathbf{x}) d\mathbf{x}$ , since the baseline density  $\rho(\mathbf{x})$  is the product of marginal

baseline densities.

Now let us discuss how to compute the integral terms  $\tilde{b}_{ij}$  separately in each dimension  $j$  with the notation  $j$  for the dimension suppressed throughout this section. For a categorical variable on the discrete domain  $\{1, \dots, L\}$ ,  $\tilde{b}_i = \sum_{x=1}^L R_1(x_i, x)\rho(x) = L\rho(x_i) - 1$ . For a continuous variable,  $\tilde{b}_i = \int_0^1 R_1(x_i, x)\rho(x)dx = \int_0^1 \{k_1(x_i)k_1(x) + k_2(x_i)k_2(x) - k_4(|x_i - x|)\}\rho(x)dx$  needs to be computed with a Beta baseline density function  $\rho(x) = \rho(x; \beta_1, \beta_2)$ . For the computation of this, let us define ascending factorial :  $x^{[r]} = x(x+1)\dots(x+r-1)$ , then, for  $y \in [0, 1]$ ,

$$l_r(y) = \int_0^y x^r \rho(x)dx = \text{betacdf}(y; \beta_1 + r, \beta_2) \frac{\beta_1^{[r]}}{(\beta_1 + \beta_2)^{[r]}},$$

$$u_r(y) = \int_y^1 x^r \rho(x)dx = \{1 - \text{betacdf}(y; \beta_1 + r, \beta_2)\} \frac{\beta_1^{[r]}}{(\beta_1 + \beta_2)^{[r]}}.$$

Letting  $m_r = \int_0^1 x^r \rho(x)dx$ , we get

$$\int_0^1 k_1(x)\rho(x)dx = m_1 - \frac{1}{2}$$

$$\int_0^1 k_2(x)\rho(x)dx = \frac{1}{2} \left( m_2 - m_1 + \frac{1}{6} \right)$$

and  $\int_0^1 k_4(|x - x_i|)\rho(x)dx$  can be computed by

$$\int_0^1 k_4(|x - y|)\rho(x)dx = \int_0^y k_4(y - x)\rho(x)dx + \int_y^1 k_4(x - y)\rho(x)dx$$

$$= \frac{1}{24} \left\{ l_4(y) + 4al_3(y) + \left( 6a^2 - \frac{1}{2} \right) l_2(y) + (4a^3 - a) l_1(y) + \left( a^4 - \frac{a^2}{2} + \frac{7}{240} \right) l_0(y) \right\}$$

$$+ \frac{1}{24} \left\{ u_4(y) + 4bu_3(y) + \left( 6b^2 - \frac{1}{2} \right) u_2(y) + (4b^3 - b) u_1(y) + \left( b^4 - \frac{b^2}{2} + \frac{7}{240} \right) u_0(y) \right\}$$

with  $a(y) = -y + 1/2$  and  $b(y) = -y - 1/2$ , since

$$k_4(y - x) = \frac{1}{24} \left\{ x^4 + 4ax^3 + \left( 6a^2 - \frac{1}{2} \right) x^2 + (4a^3 - a) x + a^4 - \frac{a^2}{2} + \frac{7}{240} \right\}$$

$$k_4(x - y) = \frac{1}{24} \left\{ x^4 + 4bx^3 + \left( 6b^2 - \frac{1}{2} \right) x^2 + (4b^3 - b) x + b^4 - \frac{b^2}{2} + \frac{7}{240} \right\}.$$

It is also possible to use other distributions than the beta distribution. For example,

when using the uniform density for  $\rho$ , the computation of  $\int g\rho$  is particularly simple and can be done analytically.

## 4.2 Newton-Raphson iteration

In this section we introduce the algorithm for finding  $g \in \mathcal{G}_n = \text{span}\{R_\theta(\mathbf{x}_i, \cdot), i = 1, \dots, n\}$  to minimize (3.3) for fixed  $\lambda$  and  $\theta = (\theta_1, \dots, \theta_p)^T$ . A function in  $\mathcal{G}_n$  has the expression  $g(\mathbf{x}) = \sum_{i=1}^n c_i R_\theta(\mathbf{x}_i, \mathbf{x}) = \sum_{\alpha=1}^p \theta_\alpha \sum_{i=1}^n c_i R_\alpha(\mathbf{x}_i, \mathbf{x})$  and its penalty  $J(g) = \sum_{\alpha=1}^p \theta_\alpha^{-1} \|P^\alpha g\|^2$  has a matrix representation  $J(g) = \mathbf{c}^T \mathbf{R}_\theta \mathbf{c}$ , where  $\mathbf{R}_\theta = \{R_\theta(\mathbf{x}_i, \mathbf{x}_j)\}_{i,j=1,\dots,n}$  and  $\mathbf{c}$  is the  $n$  column vector of coefficients with  $i$ th entry  $c_i$ . The integral term can be written as

$$\int g\rho = \sum_{\alpha=1}^p \sum_{i=1}^n \theta_\alpha c_i \int R_\alpha(\mathbf{x}_i, \mathbf{x}) \rho(\mathbf{x}) d\mathbf{x} = \theta^T \mathbf{B}^T \mathbf{c},$$

where  $\mathbf{B}$  is a  $n \times p$  matrix with  $(i, \alpha)$ th entry  $b_{i,\alpha} = \int R_\alpha(\mathbf{x}, \mathbf{x}_i) \rho(\mathbf{x}) d\mathbf{x}$ .

In the all-two-way-interaction ANOVA setting,  $\mathbf{R}_\theta = \sum_{j=1}^d \theta_j \mathbf{R}_j + \sum_{j < k} \theta_{jk} \mathbf{R}_{jk}$  where  $\mathbf{R}_j$  is the kernel matrix for the  $j$ th variable and  $\mathbf{R}_{jk}$  is the elementwise multiplication of  $\mathbf{R}_j$  and  $\mathbf{R}_k$ . The  $(i, \alpha)$ th entry  $b_{i,\alpha}$  of the  $n \times p$  matrix  $\mathbf{B}$  is  $b_{i,\alpha} = \tilde{b}_{i,j}$  for  $\alpha = 1, \dots, d$ , and  $b_{i,\alpha} = \tilde{b}_{i,j} \tilde{b}_{i,k}$  for  $\alpha = (d+1), \dots, p$ .

Denoting  $\xi_i^\theta(\cdot) = R_\theta(\mathbf{x}_i, \cdot)$  and  $\mathbf{b}_\theta = \mathbf{B}\theta$ , the minimization problem (3.3) becomes to minimize

$$A_{\lambda,\theta}(\mathbf{c}) = \log \left( \frac{1}{n} \sum_{i=1}^n \exp \left\{ - \sum_{j=1}^n c_j \xi_j^\theta(\mathbf{x}_i) \right\} \right) + \mathbf{b}_\theta^T \mathbf{c} + \lambda \mathbf{c}^T \mathbf{R}_\theta \mathbf{c}. \quad (4.1)$$

The Newton-Raphson iteration can be applied to minimize (4.1) for fixed  $\lambda$  and  $\theta$ . A direct calculation gives

$$\begin{aligned} \frac{\partial A_{\lambda,\theta}(\mathbf{c})}{\partial c_k} &= - \sum_{i=1}^n w_i(\mathbf{c}) \xi_k^\theta(\mathbf{x}_i) + \{\mathbf{b}_\theta\}_k + 2\lambda \{\mathbf{R}_\theta \mathbf{c}\}_k \\ \frac{\partial^2 A_{\lambda,\theta}(\mathbf{c})}{\partial c_k \partial c_l} &= \sum_{i=1}^n w_i(\mathbf{c}) \xi_k^\theta(\mathbf{x}_i) \xi_l^\theta(\mathbf{x}_i) - \left\{ \sum_{i=1}^n w_i(\mathbf{c}) \xi_k^\theta(\mathbf{x}_i) \right\} \left\{ \sum_{i=1}^n w_i(\mathbf{c}) \xi_l^\theta(\mathbf{x}_i) \right\} + 2\lambda \{\mathbf{R}_\theta\}_{k,l}, \end{aligned}$$

where  $\{\cdot\}_k$  denotes  $k$ th entry of the vector  $\{\cdot\}$ ,  $\{\mathbf{R}_\theta\}_{k,l}$  denotes  $(k, l)$ th entry of the matrix  $\mathbf{R}_\theta$ , and

$$w_i(\mathbf{c}) = \frac{\exp \{ -g(\mathbf{x}_i) \}}{\sum_{i=1}^n \exp \{ -g(\mathbf{x}_i) \}} = \frac{\exp \{ - \sum_{j=1}^n c_j \xi_j^\theta(\mathbf{x}_i) \}}{\sum_{i=1}^n \exp \{ - \sum_{j=1}^n c_j \xi_j^\theta(\mathbf{x}_i) \}}.$$

Letting  $\mathbf{w}_c = (w_1(\mathbf{c}), \dots, w_n(\mathbf{c}))^T$  and  $\mathbf{D}_c = \text{Diag}(w_1(\mathbf{c}), \dots, w_n(\mathbf{c}))$ , the derivatives can be expressed as

$$\begin{aligned}\frac{\partial A_{\lambda, \theta}(\mathbf{c})}{\partial \mathbf{c}} &= -\mathbf{R}_\theta \mathbf{w}_c + \mathbf{b}_\theta + 2\lambda \mathbf{R}_\theta \mathbf{c} \\ \frac{\partial^2 A_{\lambda, \theta}(\mathbf{c})}{\partial \mathbf{c} \partial \mathbf{c}^T} &= \mathbf{R}_\theta \mathbf{D}_c \mathbf{R}_\theta - (\mathbf{R}_\theta \mathbf{w}_c)(\mathbf{R}_\theta \mathbf{w}_c)^T + 2\lambda \mathbf{R}_\theta.\end{aligned}$$

Writing  $\tilde{\mathbf{c}}$  as the current iterate of  $\mathbf{c}$ , the Newton updating equation is

$$(\mathbf{R}_\theta \mathbf{D}_{\tilde{\mathbf{c}}} \mathbf{R}_\theta - \mathbf{R}_\theta \mathbf{w}_{\tilde{\mathbf{c}}} \mathbf{w}_{\tilde{\mathbf{c}}}^T \mathbf{R}_\theta + 2\lambda \mathbf{R}_\theta)(\mathbf{c} - \tilde{\mathbf{c}}) = \mathbf{R}_\theta \mathbf{w}_{\tilde{\mathbf{c}}} - \mathbf{b}_\theta - 2\lambda \mathbf{R}_\theta \tilde{\mathbf{c}}.$$

After arranging terms, we get

$$(\mathbf{R}_\theta \mathbf{D}_{\tilde{\mathbf{c}}} - \mathbf{R}_\theta \mathbf{w}_{\tilde{\mathbf{c}}} \mathbf{w}_{\tilde{\mathbf{c}}}^T + 2\lambda \mathbf{I}) \mathbf{R}_\theta \mathbf{c} = (\mathbf{R}_\theta \mathbf{D}_{\tilde{\mathbf{c}}} - \mathbf{R}_\theta \mathbf{w}_{\tilde{\mathbf{c}}} \mathbf{w}_{\tilde{\mathbf{c}}}^T) \mathbf{R}_\theta \tilde{\mathbf{c}} + \mathbf{R}_\theta \mathbf{w}_{\tilde{\mathbf{c}}} - \mathbf{b}_\theta,$$

or

$$(\mathbf{R}_\theta \mathbf{D}_{\tilde{\mathbf{c}}} - \mathbf{R}_\theta \mathbf{w}_{\tilde{\mathbf{c}}} \mathbf{w}_{\tilde{\mathbf{c}}}^T + 2\lambda \mathbf{I}) \mathbf{g} = (\mathbf{R}_\theta \mathbf{D}_{\tilde{\mathbf{c}}} - \mathbf{R}_\theta \mathbf{w}_{\tilde{\mathbf{c}}} \mathbf{w}_{\tilde{\mathbf{c}}}^T) \tilde{\mathbf{g}} + \mathbf{R}_\theta \mathbf{w}_{\tilde{\mathbf{c}}} - \mathbf{b}_\theta,$$

in terms of  $\mathbf{g} = (g(\mathbf{x}_1), \dots, g(\mathbf{x}_n))^T = \mathbf{R}_\theta \mathbf{c}$ .

### 4.3 The COSSO

We find a solution to the COSSO (3.5) among the functions of the form  $g(\mathbf{x}) = \sum_{i=1}^n c_i R_\theta(\mathbf{x}_i, \mathbf{x}) = \sum_{\alpha=1}^p \theta_\alpha \sum_{i=1}^n c_i R_\alpha(\mathbf{x}_i, \mathbf{x})$ . Now the COSSO formula can be viewed as a function of  $\theta$  and  $\mathbf{c} = (c_1, \dots, c_n)^T$ . A reasonable scheme would be to minimize (3.5) iteratively with respect to  $\theta$  and  $\mathbf{c}$ . If  $\theta$  were fixed, then, since it is a smoothing spline problem, the solution can be obtained as described in Section 4.2. On the other hand, if  $\mathbf{c}$  were fixed, denote  $\mathbf{g}_\alpha = \mathbf{R}_\alpha \mathbf{c}$ , and let  $\mathbf{G} = [\mathbf{g}_1, \dots, \mathbf{g}_p]$  be the  $n \times p$  matrix with the  $\alpha$ th column being  $\mathbf{g}_\alpha$ . So the formula (3.5) is equivalent to minimizing

$$A(\theta) = \log \left\{ \frac{1}{n} \mathbf{1}^T \exp(-\mathbf{G}\theta) \right\} + \mathbf{c}^T \mathbf{B}\theta + \lambda_0 \mathbf{c}^T \mathbf{G}\theta$$

subject to  $\theta_\alpha \geq 0$  and  $\sum \theta_\alpha \leq M$  for some  $M$ . In our algorithm, instead of  $A(\theta)$ , we propose to solve a quadratic approximation of  $A(\theta)$  for updating  $\theta$ .

Letting  $\mathbf{w}_\theta = (w_{\theta_1}, \dots, w_{\theta_n})^T$ , where  $w_{\theta_i} = \exp\{-g(\mathbf{x}_i)\} / \sum_{j=1}^n \exp\{-g(\mathbf{x}_j)\}$ , the gradient vector and the Hessian matrix of  $A(\theta)$  can be written as

$$\begin{aligned} G_A(\theta) &= -\mathbf{G}^T \mathbf{w}_\theta + \mathbf{B}^T \mathbf{c} + \lambda_0 \mathbf{G}^T \mathbf{c} \\ H_A(\theta) &= \mathbf{G}^T \{\text{Diag}(\mathbf{w}_\theta) - \mathbf{w}_\theta \mathbf{w}_\theta^T\} \mathbf{G}. \end{aligned}$$

Therefore the iteration for updating  $\theta$  is via solving the simple quadratic programming problem which minimizes the quadratic approximation of  $A(\theta)$  around current iterate  $\tilde{\theta}$

$$\begin{aligned} &A(\tilde{\theta}) + (\theta - \tilde{\theta})^T G_A(\tilde{\theta}) + \frac{1}{2}(\theta - \tilde{\theta})^T H_A(\tilde{\theta})(\theta - \tilde{\theta}) \\ &= \frac{1}{2}\theta^T H_A(\tilde{\theta})\theta + \theta^T \{G_A(\tilde{\theta}) - H_A(\tilde{\theta})\tilde{\theta}\} + \text{constant}, \end{aligned} \tag{4.2}$$

subject to  $\theta_\alpha \geq 0$  and  $\sum \theta_\alpha \leq M$ .

For fixed  $\lambda_0$  and  $M$ , our algorithm is as follows:

1. Initialization: fix  $\theta_\alpha = 1$ ,  $\alpha = 1, \dots, p$ .
2. For currently given  $\theta$ , solve for  $\mathbf{c}$ .
3. For the current  $\mathbf{c}$  and  $\tilde{\theta}$  being the current iterate of  $\theta$ , solve for  $\theta$  with (4.2).
4. repeat step 2 and step 3 until  $\theta$  converges or given number of times, whichever comes first.

#### 4.4 Choosing the smoothing parameter

Kullback-Leibler (KL) loss is often considered a measure of distance between two probability density functions. If  $\hat{p}$  is an estimate of the density function  $p$  of  $X$ , then KL loss is given by  $KL(p, \hat{p}) = E_X \log\{p(\mathbf{X})/\hat{p}(\mathbf{X})\}$ . Ignoring the term which involves only the true density  $p$ , we have the relative Kullback-Leibler loss  $RKL(p, \hat{p}) = -E_X \log \hat{p}(\mathbf{X})$ . If a test set is available, we can use the empirical relative KL loss on the test set to tune the smoothing parameter. When there is no test sample, we can use the five or ten fold cross-validation.

In computing the KL loss, we need to evaluate the constant term in (2.4). This is done through the Monte Carlo integration. Notice that this integration is required only after the iterative estimation procedure and the performance of the estimator is not very sensitive to the slight changes in the tuning parameter. Therefore the Monte Carlo integration with a reasonable sample size can do the job.

Combined with the tuning procedure, the complete algorithm to fit the COSSO estimate is as follows:

1. Fix  $\theta_\alpha = 1$ ,  $\alpha = 1, \dots, p$ . Solve the smoothing spline problem and tune  $\lambda_0$  according to CV. Set  $\lambda_0$  fixed at the chosen value in all later steps.
2. For each given  $M$  in a reasonable range, apply the COSSO algorithm with  $M$ . Tune  $M$  according to CV. The solution corresponding to this chosen  $M$  is the final solution.

## 5 Simulations

To investigate how our method performs in various problems, we generated simulated data from a number of examples. The examples considered in this section are all in the two-way interaction log-density ANOVA models.

For the univariate problem the algorithm described in Section 4.2 was directly applied. The COSSO algorithm for two-way interaction models described in Section 4.3 and Section 4.4 was applied for higher dimensional examples where the model selection is of as much interest as estimation accuracy. Our method is compared with the Gaussian kernel density estimation (GKDE) and the penalized likelihood (PL) method. The kernel density estimator used is the simplest kind which involves only one smoothing parameter. It is simple to use and works well for low dimensional problems, but expected not to work well in high dimensions. For the penalized likelihood method we used the implementation in the R library `gss` which can handle at most 4 dimensional problems. The five fold cross-validated (5-CV) log-likelihood was used in choosing smoothing parameters for our method and the GKDE. The selection of smoothing parameters for the PL method is through a modified version of the generalized approximate cross-validation (GACV) described in Gu and Wang (2003), with the default parameter value  $\alpha = 1.4$ . To evaluate the performances of the estimators

in our simulation study, we generate an independent test sample  $\{\mathbf{x}_k^*, k = 1, \dots, N\}$  from the true  $p$ , and use the empirical KL loss on the test set to compare the estimators.

*Example 5.1.* Samples of size  $n = 100$  were generated from the univariate density proportional to  $(1/3) N(.3, .1^2) + (2/3) N(.7, .1^2)$  truncated on  $[0, 1]$ . The estimated density functions by our method, the GKDE and the PL method based on the first sample are in Figure 1. For this sample, the empirical KL losses based on an independent sample of size  $N = 1000$  from the true density were  $KL_{New} = 0.0119$ ,  $KL_{GKDE} = 0.0220$  and  $KL_{PL} = 0.0103$ . We ran the simulation  $s = 100$  times more and the test set KL losses were computed in each simulation and displayed in the boxplots (Figure 1, bottom right). The boxplots show that all three methods are comparable in this one dimensional example, with the PL method performs slightly better.

*Example 5.2.* A 4 dimensional density was constructed by combining the univariate density used in Example 5.1 and a 2 dimensional density independently.  $(X_1, X_2)$  follows the 2 dimensional density proportional to

$$\frac{1}{2}N((.3, .5), I/49) + \frac{1}{3}N((.7, .7), I/49) + \frac{1}{6}N((.75, .25), I/49) \quad (5.1)$$

truncated on  $[0, 1]^2$ , and  $X_3$  and  $X_4$  follow the density in Example 5.1 independently.  $(X_1, X_2)$  is independent of  $X_3$  and  $X_4$ . Three estimation methods were applied to the  $s = 50$  samples of size  $n = 600$ .  $N = 3000$  was used for the test sample for the empirical KL.

The R function `ssden` adopt a sub-basis scheme which uses only a part of the sample as basis functions to reduce the computation. That is, the minimizer is found in the space spanned by  $\{R(\mathbf{x}_i, \cdot), i \in I, \text{ and } \phi_j(\cdot), j = 1, \dots, l\}$  for a random subset  $I$  of  $\{1, \dots, n\}$  instead of  $\{1, \dots, n\}$  itself. Let  $n_B$  denote the cardinality of the set  $I$ .

To compare the running time between our method and the PL method, we also adopted the sub-basis scheme into our method. The first sample of Example 5.2 was taken and two methods were applied with the same sets of sub-basis functions of various sizes. The total CPU time used by our MATLAB implementation according to the MATLAB function

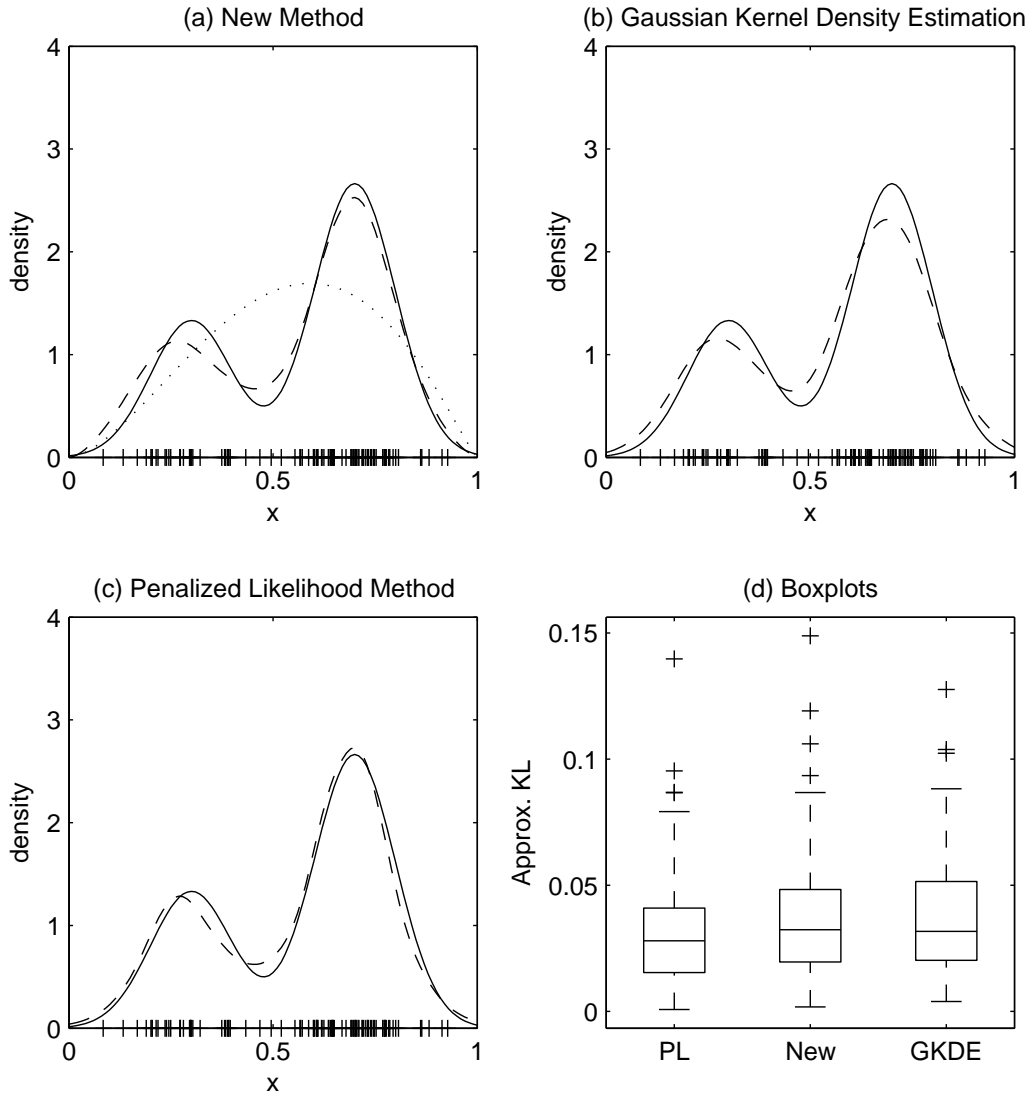


Figure 1: Based on a sample of size  $n = 100$  from the true density (solid line) of Example 5.1, the estimated densities (dashed line) by the new method (top left), the Gaussian kernel density estimation (top right), and the penalized likelihood method (bottom left) are plotted. The dotted line in the top left panel indicates baseline density. Boxplots of the test sample KL distances of estimated densities by the three methods based on  $s = 100$  simulations are plotted (bottom right). The smoothing parameters were selected through 5-CV for the new method and GKDE, and a modified GACV for the penalized likelihood method.

$n_B$	our method	PL
42	120.50	3435.79
100	213.05	9418.51
200	427.10	Fail
600	1828.08	Fail

Table 1: Total elapsed times (in seconds) for our method and the R `ssden` process.

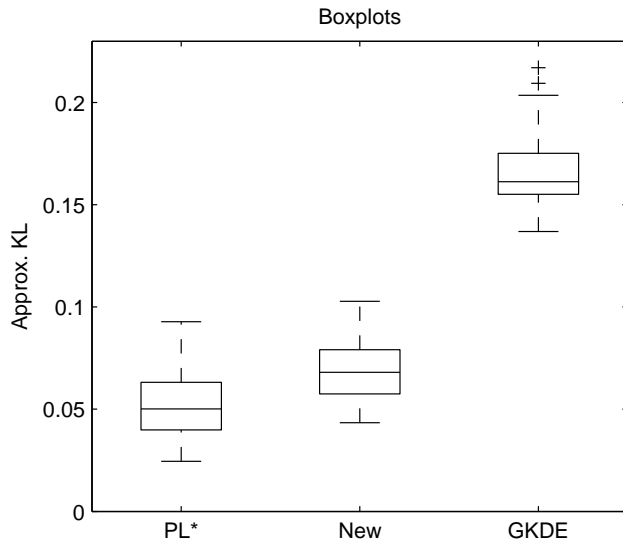


Figure 2: Example 5.2. Boxplots of the test set KL distances of estimated densities by our method and the GKDE based on  $s = 50$  simulations. The boxplot for the PL method is based on only the successful 47 runs.

`cputime`, and the total elapsed times for the R process by the function `ssden` according to the R function `proc.time` are presented for  $n_B = 42, 100, 200, 600$  in Table 1, where  $n_B = 42$  is the default option of the `ssden` function. The PL method failed with  $n_B = 200, 300, 400, 500, 600$ .

For the simulation of Example 5.2, we applied our method and the `ssden` function with  $n_B = 42$ . The same basis functions were used in each simulation. Another difficulty arose in the `ssden` function: the procedure sometimes failed to fit the model. Although `ssden` returned answers in all simulations, the warnings “Newton iteration fails to converge” followed in the 3 cases out of total  $s = 50$  simulations, where the answers were far from reasonable estimates. The algorithm finished successfully in the rest 47 cases.

Figure 2 shows boxplots of the test set KL distances for our method and the GKDE

based on  $s = 50$  simulations. In addition to the boxplots for our method and the GKDE, the boxplot for the PL method based on only the 47 cases where `ssden` was successful is also displayed in Figure 2. It is clear that our method performs much better than the GKDE. If we only use the 47 cases where the `ssden` function successfully obtained the estimates to compute the error for the PL method, the PL method performed better than our method. Notice this computed error is biased in favor of the PL method since we have effectively deleted the outliers in the performance measure of the PL method.

In higher dimensional problems, it is appropriate to use much more basis functions, and the computational cost for integration also increases exponentially. Therefore the PL method is not suitable for high dimensional problems. The implementation in `gss` can handle at most 4 dimensional problems. Our method is computationally much more practical. For the following examples with dimension higher than 4, the comparison is made only between our method and GKDE since the PL method can not be computed.

*Example 5.3.* Samples of size  $n = 600$  were generated from the 5-variate normal density truncated on  $[0, 1]^5$  with mean  $(.5, .5, .5, .5, .5)$  and the covariance matrix  $\Sigma_1$  with

$$\Sigma_1^{-1} = \begin{pmatrix} 62 & -30 & 0 & 0 & -30 \\ -30 & 62 & -15 & 0 & 0 \\ 0 & -15 & 62 & 13 & 0 \\ 0 & 0 & 13 & 62 & -19 \\ -30 & 0 & 0 & -19 & 62 \end{pmatrix}.$$

Notice that  $X_j \perp\!\!\!\perp X_k \mid (\text{the rest})$  for  $(j, k) = (1, 3), (1, 4), (2, 4), (2, 5), (3, 5)$  so the corresponding graph is a chordless 5-cycle (Figure 3, left panel). We repeat the simulation  $s = 50$  times and the test set size is  $N = 3000$ .

*Example 5.4.* Another 5 dimensional density was constructed by combining two 2 dimensional densities and the univariate density used in Example 5.1 independently.  $(X_1, X_2)$  follows the 2 dimensional density proportional to (5.1) truncated on  $[0, 1]^2$ ,  $X_5$  follows Ex-

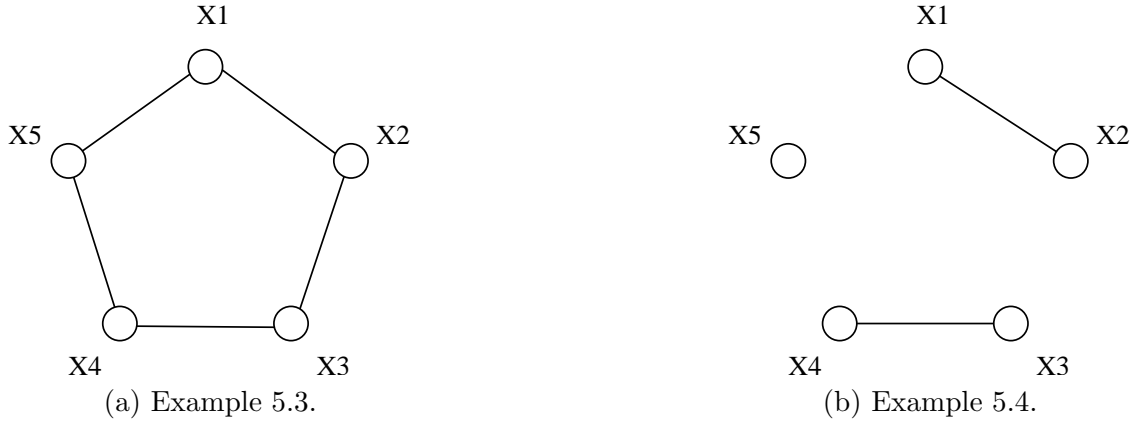


Figure 3: Graphs for two 5-D examples. The edges indicate the interaction terms present in the true density.

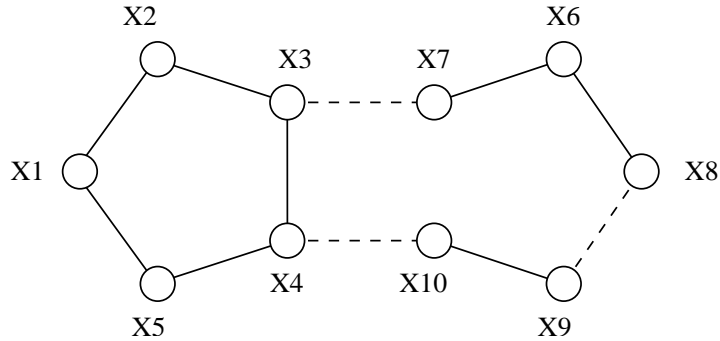


Figure 4: Graph for Example 5.5. The interactions corresponding to the solid edges are those present in the true density. The interactions corresponding to the dashed edges are additionally included in the estimation procedure.

ample 5.1, and  $(X_3, X_4)$  follows the 2 dimensional density

$$\frac{2}{3}\text{Beta}(2, 4)^2 + \frac{1}{3}\text{Beta}(7, 4)^2, \quad (5.2)$$

where  $\text{Beta}(2, 4)^2$  represents the 2 dimensional distribution where each variable independently follows  $\text{Beta}(2, 4)$  and  $\text{Beta}(7, 4)^2$  is defined similarly.  $(X_1, X_2)$ ,  $(X_3, X_4)$ , and  $X_5$  are independent of each other. Figure 3 (right panel) shows the corresponding graph. Two estimation methods were applied to the  $s = 50$  samples of size  $n = 600$ . The test set size is  $N = 3000$ .

*Example 5.5.* A 10 dimensional density was constructed by independently combining  $(X_1, \dots, X_5)$

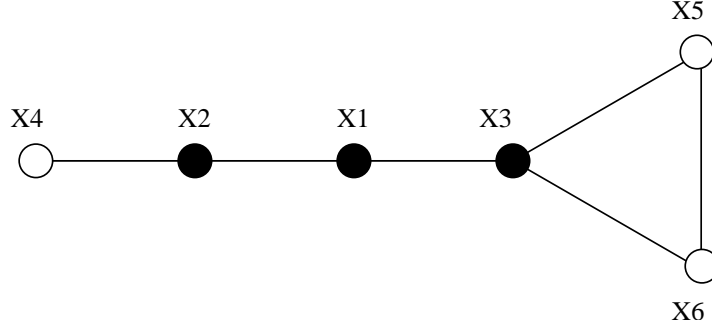


Figure 5: Graph for Example 5.6. Continuous variables are indicated by the circle ( $\circ$ ) vertices and discrete variables by the dot ( $\bullet$ ).

from Example 5.3,  $(X_6, X_7, X_8)$  from the 3-variate normal density truncated on  $[0, 1]^3$  with mean  $(.5, .5, .5)$  and the covariance matrix  $\Sigma_2$  with

$$\Sigma_2^{-1} = \frac{1}{1.2} \begin{pmatrix} 62 & -30 & -30 \\ -30 & 62 & 0 \\ -30 & 0 & 62 \end{pmatrix},$$

and  $(X_9, X_{10})$  from (5.2). Notice  $X_7 \perp\!\!\!\perp X_8 \mid X_6$ . We considered only 11 two-way interactions which include the 8 interactions present in the true density (Figure 4). With sample size  $n = 600$ , the simulation was repeated  $s = 50$  times.  $N = 5000$  was used.

*Example 5.6.* A mixed model was constructed in the following way:

$(X_1, X_2, X_3)$  is drawn from the 3-variate discrete distribution with probability  $P(X_1 = i, X_2 = j, X_3 = k) = p_{ijk}$ ,  $i, j = 1, 2$ ,  $k = 1, 2, 3$ , where

$$\begin{aligned} p_{111} &= 0.10010 & p_{112} &= 0.05839 & p_{113} &= 0.10219 \\ p_{211} &= 0.06239 & p_{212} &= 0.08592 & p_{213} &= 0.05155 \\ p_{121} &= 0.07369 & p_{122} &= 0.04298 & p_{123} &= 0.07522 \\ p_{221} &= 0.10850 & p_{222} &= 0.14942 & p_{223} &= 0.08965 \end{aligned}$$

We remark  $X_2 \perp\!\!\!\perp X_3 \mid X_1$ . Conditional on  $(X_1, X_2, X_3)$ ,  $X_4$  follows  $p(x_4|x_1, x_2, x_3) = p(x_4|x_2) = p_1(x_4)I(x_2 = 1) + p_2(x_4)I(x_2 = 2)$  and  $(X_5, X_6)$  follows  $p(x_5, x_6|x_1, x_2, x_3) = p(x_5, x_6|x_3) = q(x_5, x_6, x_3)$ . Here  $p_1$  is the density function of Example 5.1,  $p_2$  is the density

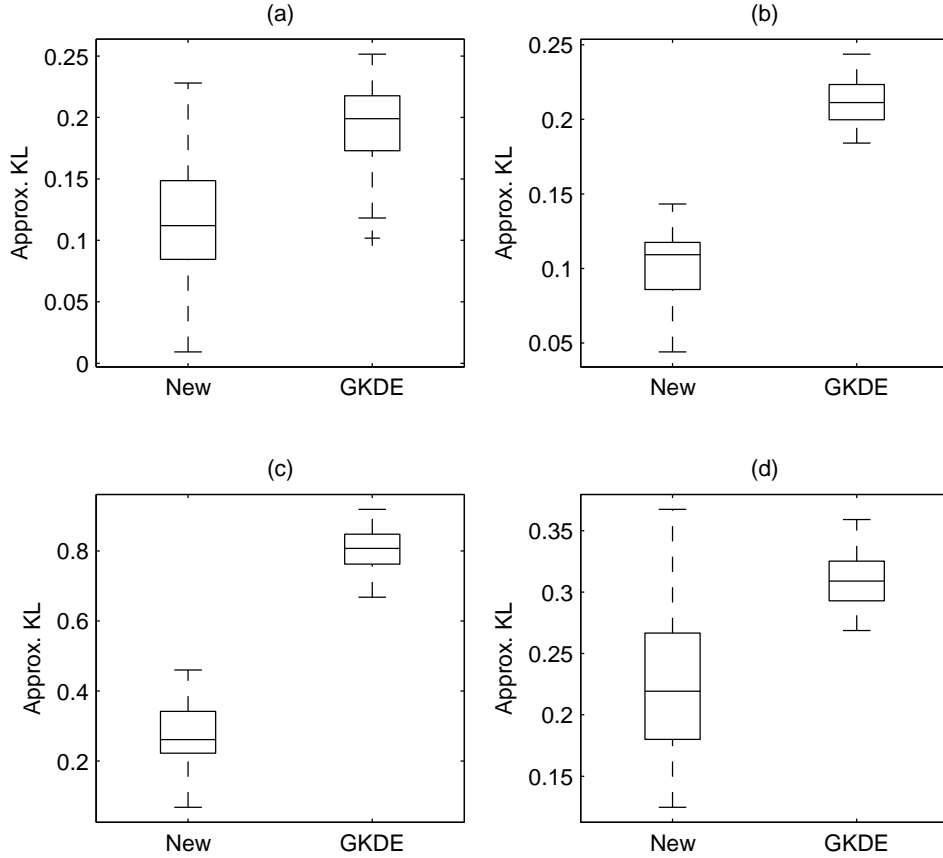


Figure 6: Boxplots of the test set KL distances of estimated densities by two methods in (a) Example 5.3, (b) Example 5.4, (c) Example 5.5, and (d) Example 5.6 based on  $s = 50$  simulations.

function of  $0.6\text{Beta}(12, 7) + 0.4\text{Beta}(3, 11)$  on  $[0, 1]$ , and  $q(u, v, w)$  is proportional to

$$\exp\{\log r_1(u, v) + \log r_2(u) - I(w = 1) \log r_2(u) + I(w = 2) \log r_3(v) + I(w = 3) \log r_2(v)\}$$

with  $r_1(u, v)$  being the density of  $N\left((.5, .5), \begin{pmatrix} 62 & -30 \\ -30 & 62 \end{pmatrix}^{-1}\right)$ ,  $r_2(u)$  the density of  $0.5\text{Beta}(12, 3) + 0.5\text{Beta}(3, 12)$ , and  $r_3(u)$  the density of  $\text{Beta}(3, 5)$ . Notice that  $\log p(x_1, x_2, x_3, x_4, x_5, x_6) = \log p(x_1, x_2, x_3) + \log p(x_4|x_1, x_2, x_3) + \log p(x_5, x_6|x_1, x_2, x_3)$  has the form  $\log p = \eta_{12} + \eta_{13} + \eta_{24} + \eta_{35} + \eta_{36} + \eta_{56}$  and the corresponding graph is shown in Figure 5. With sample size  $n = 600$ , the simulation was repeated  $s = 50$  times.  $N = 3000$  was used.

For the Examples 5.3, 5.4, 5.5, and 5.6, boxplots of the test set KL distances for our method and the GKDE based on  $s = 50$  simulations are displayed in Figure 6. It is clear

Interactions	12	13	14	23	24	34									
example 5.2	1	0	0	0	0	0									
	50	19	19	23	22	17									
Interactions	12	13	14	15	23	24	25	34	35	45					
example 5.3	1	0	0	1	1	0	0	1	0	1					
	50	11	9	50	49	7	11	40	6	50					
example 5.4	1	0	0	0	0	0	0	1	0	0					
	50	24	20	19	28	20	14	50	18	18					
Interactions	12	15	23	34	37	45	4T	67	68	89	9T				
example 5.5	1	1	1	1	0	1	0	1	1	0	1				
	50	50	37	21	1	49	0	50	50	3	50				
Interactions	12	13	14	15	16	23	24	25	26	34	35	36	45	46	56
example 5.6	1	1	0	0	0	0	1	0	0	0	1	1	0	0	1
	50	50	14	9	10	9	49	9	13	15	50	50	5	3	50

Table 2: The frequency of appearance of the two-way interactions in the selected models in 50 runs. The numbers in the interaction row represent the corresponding variable, for instance  $9T$  represents the interaction between  $X_9$  and  $X_{10}$ . In each example, the existence of a term in the true density is indicated by 1 and absence by 0 on top, and the counted frequency is on bottom.

that our method performs better than the GKDE and much better in Example 5.4 and 5.5. It appears that the performance of GKDE deteriorates fast as the dimension increases while our method does not.

To study the model selection performance of our method, the number of times each component appears in the  $s = 50$  chosen models was determined for the high dimensional examples. In our computation we regarded a  $\theta$  as zero if it is smaller than  $10^{-15}$ . Notice that we have a hierarchical structure in the selected model due to the baseline density, that is, main effects are always included in the selected model. Hence, the number of times each two-way interaction appears in the chosen models was counted and is shown in Table 2. The numbers in the interaction row represent the corresponding variable, for instance  $9T$  represents the interaction term  $\eta_{9T}$  between  $X_9$  and  $X_{10}$  (denoted by T). In the first row of each example, 1 indicates the existence of the corresponding term in the true density and 0 indicates the absence. Our method never missed the interaction present in the true model in Example 5.2 and Example 5.4. In Example 5.3, the interaction  $\eta_{34}$  was missed 10 times, but the interaction  $\eta_{23}$  was missed only once and all the other correct interactions were never

missed. The interactions  $\eta_{23}$  and  $\eta_{34}$  were missed quite often in Example 5.5, however the overall selection is pretty good in this example considering false terms were rarely selected. The present interactions in the true density of Example 5.6 were selected all the time except once. The interactions absent in the true density are also selected sometimes, about 10% of the time. In general, we notice that the correct interactions are detected very well by our method, but our method tends to include some false terms in the chosen model. This is probably related to that we are choosing the tuning parameter according to estimation accuracy rather than model selection.

## 6 Real examples

As a practical application of our method, we consider two real examples. We apply our method to one continuous example and one mixed example. Example 6.1 is taken from Edwards (2000). The data of Example 6.2 is available at the Repository of Machine Learning Database at the University of California, Irvine. The URL is [www.ics.uci.edu/~mlearn/MLRepository.html](http://www.ics.uci.edu/~mlearn/MLRepository.html). For the analysis of the data with our method, all the continuous variables are scaled so that the values of each dimension fall in the unit interval  $[0, 1]$ .

*Example 6.1.* [Mathematics Marks] This data originally comes from Mardia et al. (1979). Examination grades of 88 students on five subjects are measured on the scale of 0–100. The subjects are mechanics, vectors, algebra, analysis, and statistics. The selected model by our method is in Figure 7. It states that, given Algebra, Analysis and Statistics are conditionally independent of Mechanics and Vectors, and the dependence of the Vectors on the others is indirect through Mechanics. This result is different from the analysis by the the traditional Graphical Gaussian models which assume that data follow a Gaussian distribution. The pairwise scatter plots (Figure 8) indicates that it may not be appropriate to assume normality.

*Example 6.2.* [Bupa Liver Disorder] This data originated from the classification problem contributed by R. S. Forsyth. The data set consists of the record of 345 male individuals on six continuous explanatory variables and a selector. Six explanatory variables consist of

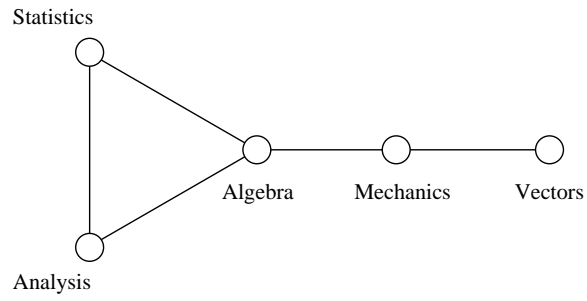


Figure 7: The selected model of Mathematics Marks.

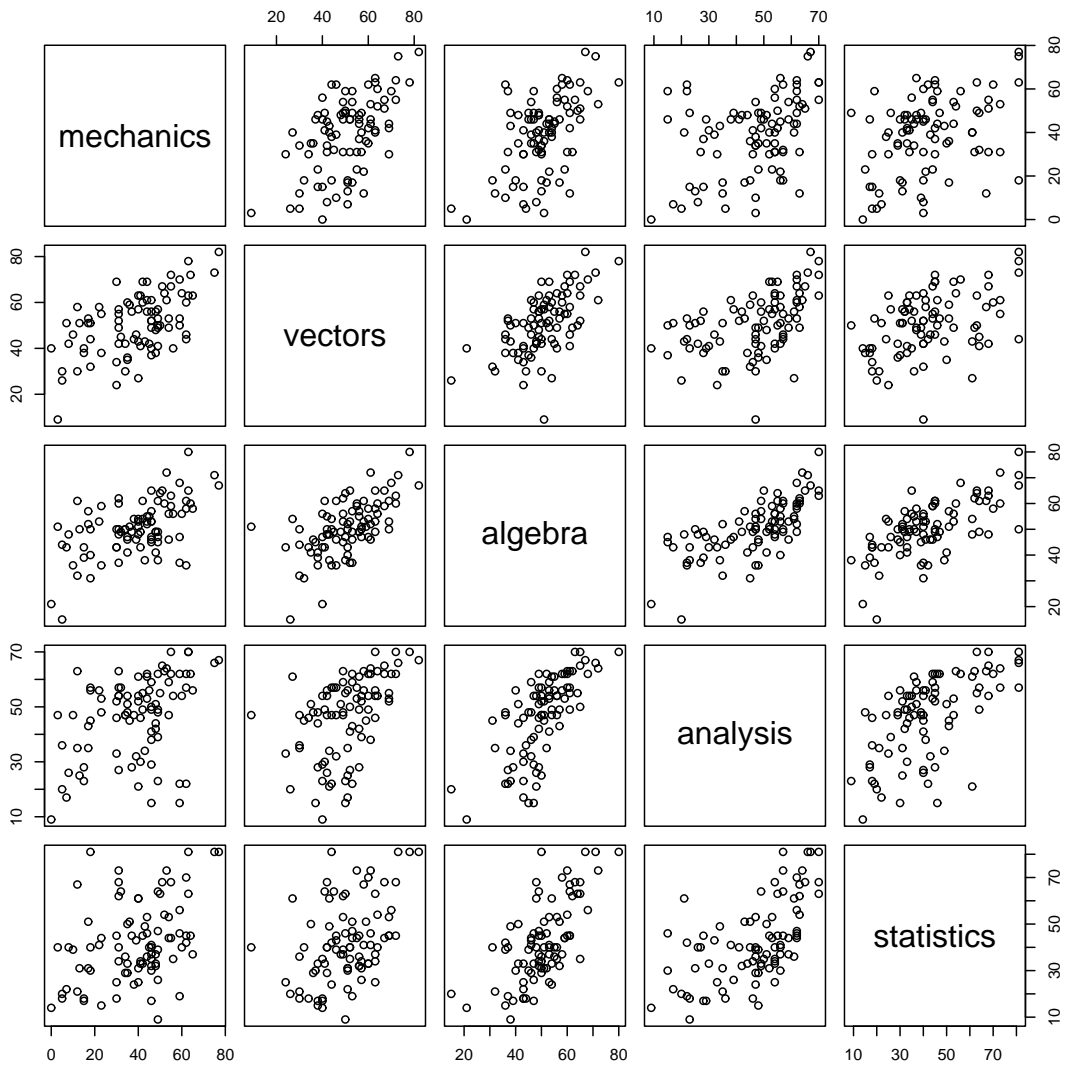


Figure 8: Mathematics marks

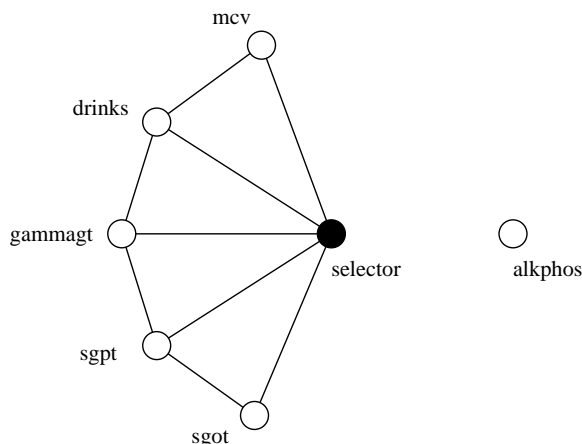


Figure 9: The selected model of Bupa Liver Disorder

**drinks** which measures the number of half-pint equivalents of alcoholic beverages drunk per day and five blood tests variables (**mcv**, **alkphos**, **sgpt**, **sgot**, **gammagt**) which are thought to be sensitive to liver disorders that might arise from excessive alcohol consumption. The graph of the selected model by our method is shown in Figure 9. It states that (i) **alkphos** is irrelevant to all the other variables, (ii) all the explanatory variables except **alkphos** have direct effects on the disorders, and (iii) **drinks** does not have direct effects on **sgpt**, **sgot**, and **alkphos**.

## 7 Discussion

A different choice of the baseline density  $\rho$  yields a different path of estimates from extremely rough functions to the density function  $\rho$ , since the limiting estimate goes to  $\rho$  as the amount of smoothing increases. It would be interesting to investigate how the estimator behaves for the different choices of  $\rho$  in situations.

Another important question is the choice of the smoothing parameters. One possibility is to reserve an independent test set for the tuning. Another possibility is the  $k$ -fold cross validation, as used in this paper. It is hopeful that some easily computable approximation to the leave out one cross validation can be developed for log-density estimation, so that we do not have to reserve an independent tuning set.

## References

- COX, D. D. and O’SULLIVAN, F. (1990). Asymptotic analysis of penalized likelihood and related estimators. *The Annals of Statistics* **18** 1676–1695.
- EDWARDS, D. (2000). *Introduction to Graphical Modelling*. Springer-Verlag Inc.
- GU, C. (1993). Smoothing spline density estimation: A dimensionless automatic algorithm. *Journal of the American Statistical Association* **88** 495–504.
- GU, C. (2002). *Smoothing Spline ANOVA Models*. Springer-Verlag Inc.
- GU, C. and QIU, C. (1993). Smoothing spline density estimation: Theory. *The Annals of Statistics* **21** 217–234.
- GU, C. and WANG, J. (2003). Penalized likelihood density estimation: Direct cross-validation and scalable approximation. *Statistica Sinica* **13** 811–826.
- LEONARD, T. (1978). Density estimation, stochastic processes and prior information. *Journal of the Royal Statistical Society, Series B, Methodological* **40** 113–132.
- LIN, Y. and ZHANG, H. H. (2002). Component selection and smoothing in smoothing spline analysis of variance models. Tech. Rep. 1072, Department of Statistics, University of Wisconsin–Madison.
- MARDIA, K. V., KENT, J. T. and BIBBY, J. M. (1979). *Multivariate Analysis*. Academic Press: London.
- SILVERMAN, B. W. (1982). On the estimation of a probability density function by the maximum penalized likelihood method. *The Annals of Statistics* **10** 795–810.
- WAHBA, G. (1990). *Spline Models for Observational Data*. SIAM, Philadelphia.
- WHITTAKER, J. (1990). *Graphical Models in Applied Multivariate Statistics*. John Wiley & Sons.