Statistica Sinica **16**(2006), 353-374

AN EFFECTIVE METHOD FOR HIGH-DIMENSIONAL LOG-DENSITY ANOVA ESTIMATION, WITH APPLICATION TO NONPARAMETRIC GRAPHICAL MODEL BUILDING

Yongho Jeon and Yi Lin

University of Wisconsin

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 practical 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.

Key words and phrases: Density estimation, functional ANOVA model, graphical model, model selection, penalized M-estimation.

1. Introduction

Consider the density estimation problem, in which we are given a random sample of a d-dimensional random vector $X = (X_1, \ldots, X_d)$ and 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 (analysis of variance) 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:

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

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 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 (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 η_{23} and η_{123} in the log-density ANOVA decomposition (1) indicates that the random variables X_2 and X_3 are conditionally independent given X_1 (in symbols, $X_2 \perp\!\!\!\perp X_3 \mid X_1$). 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 log-linear models; when the variables are continuous, the current research on graphical models 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 is 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 all-two-way-interaction log-linear 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

$$\arg\min_{\eta} \Big\{ -\frac{1}{n}\sum_{i=1}^{n} \eta(\mathbf{x}_i) + \int e^{\eta} + \lambda J(\eta) \Big\} \tag{2}$$

over a reproducing kernel Hilbert space $\mathcal{H}$, where J is a penalty functional that involves only derivatives. 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

$$\arg\min_{\eta} \Big\{ -\frac{1}{n}\sum_{i=1}^{n} \eta(\mathbf{x}_i) + \log \int e^{\eta} + \lambda J(\eta) \Big\}. \tag{3}$$

Gu (1993) provided a practical algorithm for (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 η is in an all-two-way interaction model. For each fixed tuning parameter, solving (2) or (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 accuracy of the approximation is required for successful Newton-Raphson iterations. A sparse grid method has been used in Gu and Wang (2003) for the integration. However, it still cannot provide sufficient accuracy with a 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 it 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 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 $p(\mathbf{x})$ of X and ρ be a fixed positive density function over $\mathcal{X}$. We propose to find $f \in \mathcal{H}$ which minimizes

$$l_n(f) + \lambda J(f), \tag{4}$$

where

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

Here $\mathcal{H}$ is a reproducing kernel Hilbert space (typically a Sobolev Hilbert space or a tensor product 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 (4) 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 (4) converges to the minimizer of the population version of l_n, when the tuning parameter λ is chosen to go to zero at a certain rate. The population version of the l_n in our case is

$$l(f) = E[e^{-f(\mathbf{X})}] + \int f(\mathbf{x})\rho(\mathbf{x})d\mathbf{x} = \int e^{-f(\mathbf{x})}p(\mathbf{x})d\mathbf{x} + \int f(\mathbf{x})\rho(\mathbf{x})d\mathbf{x}. \quad (5)$$

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

$$Dl(f)h = -\int e^{-f(\mathbf{x})}h(\mathbf{x})p(\mathbf{x})d\mathbf{x} + \int h(\mathbf{x})\rho(\mathbf{x})d\mathbf{x},$$
$$D^2l(f)gh = \int e^{-f(\mathbf{x})}g(\mathbf{x})h(\mathbf{x})p(\mathbf{x})d\mathbf{x},$$

where D denotes Fréchet derivative operator. By setting $Dl(f)h = \int h(\mathbf{x})[\rho(\mathbf{x}) - \exp\{-f(\mathbf{x})\}p(\mathbf{x})]d\mathbf{x}$ to zero for all $h \in \mathcal{H}$, we get $\rho - \exp(-f)p = 0$, that is, $\exp(f)\rho = p$. Also, l is strictly convex since $D^2l(f)gg = \int \exp\{-f(\mathbf{x})\}g(\mathbf{x})^2p(\mathbf{x})$ $d\mathbf{x} > 0$ for any nonzero $g \in \mathcal{H}$. Therefore $l(f)$ is uniquely minimized by $\bar{f} = \log p - \log \rho$, 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. If $\hat{f}$ is the solution to the minimizing problem (4), the estimate of the density p will be proportional to $\exp(\hat{f})\rho$.

The function $\rho(\mathbf{x})$ is a *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 $p(\mathbf{x})$. The optimization problem (4) involves an integral $\int f(\mathbf{x})\rho(\mathbf{x})d\mathbf{x}$. As we will see in Section 4.1, with suitable choices of ρ, 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 $\int f(\mathbf{x})\rho(\mathbf{x})d\mathbf{x}$ 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 (4) over $f \in \mathcal{H}$ is equivalent to finding $g \in \mathcal{G}$ which minimizes

$$\log\left\{\frac{1}{n}\sum_{i=1}^{n} e^{-g(\mathbf{x}_i)}\right\} + \int g(\mathbf{x})\rho(\mathbf{x})d\mathbf{x} + \lambda J(g), \tag{6}$$

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

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

If $\hat{g}$ minimizes (6), then the estimator for the density is of the form

$$\hat{p}(\mathbf{x}) = \text{constant } e^{\hat{g}(\mathbf{x})}\rho(\mathbf{x}). \tag{7}$$

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 (6), the normalizing constant in (7) requires a high-dimensional integration. However, this step is separate from the optimization procedure, and is only needed once. This does not cause serious computational problems.

3. Function space of smoothing spline ANOVA

Let $\mathcal{H}^{(j)}$ be a reproducing kernel Hilbert space 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 a reproducing kernel Hilbert space of functions on $\mathcal{X} = \mathcal{X}_1 \times \cdots \times \mathcal{X}_d$ through the tensor product space strategy:

$$\bigotimes_{j=1}^{d}\mathcal{H}^{(j)} = \bigotimes_{j=1}^{d}\left\{\{1^{(j)}\} \oplus \mathcal{H}_1^{(j)}\right\} = \{1\} \oplus \left\{\bigoplus_{j=1}^{d}\mathcal{H}_1^{(j)}\right\} \oplus \left\{\bigoplus_{j<k}[\mathcal{H}_1^{(j)} \otimes \mathcal{H}_1^{(k)}]\right\} \oplus \cdots, \tag{8}$$

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.

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

$$< f_1, f_2 >= \int_0^1 f_1(t)dt \int_0^1 f_2(t)dt + \int_0^1 f_1'(t)dt \int_0^1 f_2'(t)dt + \int_0^1 f_1''(t)f_2''(t)dt,$$

$\mathcal{H}^{(j)}$ is a reproducing kernel Hilbert space with the reproducing kernel $R(s,t) = 1 + R_1(s,t) = 1 + k_1(s)k_1(t) + k_2(s)k_2(t) - k_4(|s-t|)$, where

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

and R_1 is the reproducing kernel of $\mathcal{H}_1^{(j)}$. See Wahba (1990) and Gu (2002).

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 ρ such that $\log \rho$ has an additive structure, as we will do in our implementation, then $f = \log p - \log \rho$ and the log-density share the same ANOVA structure. In the smoothing spline ANOVA model, we assume each functional component in the decomposition (1) of f lies in a corresponding subspace in the orthogonal decomposition (8) of $\bigotimes_{j=1}^d \mathcal{H}^{(j)}$. Thus the function space $\mathcal{H}$ assumed for f consists of some orthogonal component subspaces in (8). Relabeling the subspaces other than the null space $\mathcal{N} = \{1\}$ in the model as $\mathcal{G}^{(\alpha)}$, $\alpha = 1, \ldots, p$, then $\mathcal{H} = \mathcal{N} \oplus \{\bigoplus_{\alpha=1}^p \mathcal{G}^{(\alpha)}\}$, and the smoothing spline method finds $f \in \mathcal{H}$ to minimize

$$l_n(f) + \lambda \sum_{\alpha=1}^p \theta_\alpha^{-1} \|P^\alpha f\|^2, \tag{9}$$

where P^α is the orthogonal projector in $\mathcal{H}$ into $\mathcal{G}^{(\alpha)}$. Here $\theta_\alpha \geq 0$ and if $\theta_\alpha = 0$, 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 (9) in the regression problem is in the finite dimensional space $\mathcal{N} \oplus \mathcal{G}_n$, where $\mathcal{G}_n = \text{span}\{R_\theta(\mathbf{x}_i, \cdot), i = 1, \ldots, n\}$ and $R_\theta(s,t) = \sum_{\alpha=1}^p \theta_\alpha R_\alpha(s,t)$ (Wahba (1990, Chap. 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 (4) with $J(f) = \sum_{\alpha=1}^{p} \theta_\alpha^{-1} \|P^\alpha f\|^2$, and it is equivalent to finding $g \in \mathcal{G}_n$ to minimize

$$\log\Big\{\frac{1}{n}\sum_{i=1}^{n} e^{-g(\mathbf{x}_i)}\Big\} + \int g(\mathbf{x})\rho(\mathbf{x})d\mathbf{x} + \lambda \sum_{\alpha=1}^{p} \theta_\alpha^{-1} \|P^\alpha g\|^2. \tag{10}$$

Note that the Representer Theorem for smoothing spline regression does not hold for our estimator, and we seek a good approximate solution in the finite-dimensional space $\{1\} \oplus \mathcal{G}_n$.

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

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

$$\log\Big\{\frac{1}{n}\sum_{i=1}^{n} e^{-g(\mathbf{x}_i)}\Big\} + \int g(\mathbf{x})\rho(\mathbf{x})d\mathbf{x} + \tau \sum_{\alpha=1}^{p} \|P^\alpha g\|. \tag{11}$$

It is easy to show, with arguments similar to those in Lin and Zhang (2002), that the minimization problem (11) is equivalent to the problem of finding $\theta = (\theta_1, \ldots, \theta_p)^T$ and $g \in \mathcal{G}_n$ to minimize

$$\log\Big\{\frac{1}{n}\sum_{i=1}^{n} e^{-g(\mathbf{x}_i)}\Big\} + \int g(\mathbf{x})\rho(\mathbf{x})d\mathbf{x} + \lambda_0 \sum_{\alpha=1}^{p} \theta_\alpha^{-1} ||P^\alpha g||^2 + \lambda \sum_{\alpha=1}^{p} \theta_\alpha, \tag{12}$$

subject to $\theta_\alpha \geq 0$, $\alpha = 1, \ldots, p$, where λ_0 is a fixed constant and λ is a smoothing parameter. Note that there is only one smoothing parameter λ in (12). The θ_α are not free smoothing parameters but part of the estimate.

For any fixed θ, the COSSO (12) is a smoothing spline problem (10) with fixed smoothing parameters. Thus we find a solution to (12) 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 log-density all-two-way-interaction model in our implementation. For the solution $\hat{g}$ in the all-two-way-interaction model, the estimated log-density $\hat{p}(\mathbf{x}) = \text{constant} + \hat{g}(\mathbf{x}) + \log \rho(\mathbf{x})$ preserves the two-way interaction structure if $\log \rho$ is additive.

4.1. Baseline density function

We assume the domain of each variable is $[0,1]$, and use the Beta family for the baseline density. We fit a Beta density to the marginal distribution of X_j with the method of maximum likelihood. Let $\rho^{(j)}$ be the fitted density function. Then the product of marginal baseline densities $\rho^{(j)}$ serves as the baseline density ρ.

In the log-density all-two-way-interaction model, the solution $g = \sum_{j=1}^d g_j + \sum_{j<k} g_{jk}$ to the problem of minimizing (10) is to be found among the functions of the form

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

where $R_j(\mathbf{x}, \mathbf{x}') = R_j(x_j, x_j')$ for $j = 1, \ldots, d$. The integral term in (10) can be computed as

$$\int g(\mathbf{x})\rho(\mathbf{x})d\mathbf{x} = \sum_{i=1}^n c_i \Big\{ \sum_{j=1}^d \theta_j \tilde{b}_{ij} + \sum_{j<k} \theta_{jk} \tilde{b}_{ij} \tilde{b}_{ik} \Big\},$$

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. Here $\tilde{b}_i = \int_0^1 R_1(x_i, x)\rho(x)dx = \int_0^1 \{k_1(x_i)k_1(x) + k_2(x_i)k_2(x) - k_4(|x_i - x|)\}\rho(x)dx$ needs to be computed with a Beta baseline density function $\rho(x) = \rho(x; \beta_1, \beta_2)$. For this computation, let $x^{[r]} = x(x+1)\cdots(x+r-1)$. 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

$$\int_0^1 k_1(x)\rho(x)dx = m_1 - \frac{1}{2}$$
$$\int_0^1 k_2(x)\rho(x)dx = \frac{1}{2}\Big(m_2 - m_1 + \frac{1}{6}\Big)$$

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

$$\int_0^1 k_4(|x-y|)\rho(x)dx = \int_0^y k_4(y-x)\rho(x)dx + \int_y^1 k_4(x-y)\rho(x)dx$$
$$= \frac{1}{24}\left\{l_4(y)+4al_3(y)+\left(6a^2-\frac{1}{2}\right)l_2(y)+\left(4a^3-a\right)l_1(y)+\left(a^4-\frac{a^2}{2}+\frac{7}{240}\right)l_0(y)\right\}$$
$$+\frac{1}{24}\left\{u_4(y)+4bu_3(y)+\left(6b^2-\frac{1}{2}\right)u_2(y)+\left(4b^3-b\right)u_1(y)+\left(b^4-\frac{b^2}{2}+\frac{7}{240}\right)u_0(y)\right\}$$

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

It is also possible to use other distributions than the beta distribution. For example, when using the uniform density for ρ, 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 $g \in \mathcal{G}_n = \text{span}\{R_\theta(\mathbf{x}_i, \cdot), i = 1, \ldots, n\}$ to minimize (10) for fixed λ and $\theta = (\theta_1, \ldots, \theta_p)^T$. 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 $\mathbf{R}_\theta = \{R_\theta(\mathbf{x}_i, \mathbf{x}_j)\}_{i,j=1,\ldots,n}$ and $\mathbf{c}$ is the n column vector of coefficients with ith entry c_i. The integral term can be written as

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

where $\mathbf{B}$ is a $n \times p$ matrix with (i, α)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, $\mathbf{R}_\theta = \sum_{j=1}^d \theta_j \mathbf{R}_j + \sum_{j<k} \theta_{jk} (\mathbf{R}_j \circ \mathbf{R}_k)$, where $\mathbf{R}_j$ is the kernel matrix for the jth variable and $\circ$ is used for the element-wise matrix product operator. The (i, α)th entry $b_{i,\alpha}$ of the $n \times p$ matrix $\mathbf{B}$ is $b_{i,\alpha} = \tilde{b}_{ij}$ for $\alpha = 1, \ldots, d$, and $b_{i,\alpha} = \tilde{b}_{ij}\tilde{b}_{ik}$ for $\alpha = (d+1), \ldots, p$.

Denoting $\xi_i^\theta(\cdot) = R_\theta(\mathbf{x}_i, \cdot)$ and $\mathbf{b}_\theta = \mathbf{B}\theta$, the minimization problem (10) is

$$A_{\lambda,\theta}(\mathbf{c}) = \log\left(\frac{1}{n}\sum_{i=1}^n \exp\left\{-\sum_{j=1}^n c_j \xi_j^\theta(\mathbf{x}_i)\right\}\right) + \mathbf{b}_\theta^T \mathbf{c} + \lambda \mathbf{c}^T \mathbf{R}_\theta \mathbf{c}. \tag{13}$$

The Newton-Raphson iteration can be applied to minimize (13) for fixed λ and θ. A direct calculation gives

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

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

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

Letting $\mathbf{w}_\mathbf{c} = (w_1(\mathbf{c}), \ldots, w_n(\mathbf{c}))^T$ and $\mathbf{D}_\mathbf{c} = \mathrm{Diag}(w_1(\mathbf{c}), \ldots, w_n(\mathbf{c}))$, the gradient vector and the Hessian matrix of $A_{\lambda,\theta}(\mathbf{c})$ can be written as

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

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

$$G_{A_{\lambda,\theta}}(\tilde{\mathbf{c}}) + H_{A_{\lambda,\theta}}(\tilde{\mathbf{c}})(\mathbf{c} - \tilde{\mathbf{c}}) = 0.$$

After arranging terms we get, successively,

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

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

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

4.3. The COSSO

We find a solution to the COSSO (12) among the functions of the form $g(\mathbf{x}) = \sum_{i=1}^n c_i R_\theta(\mathbf{x}_i, \mathbf{x}) = \sum_{\alpha=1}^p \theta_\alpha \sum_{i=1}^n c_i R_\alpha(\mathbf{x}_i, \mathbf{x})$. The COSSO formula can be viewed as a function of θ and $\mathbf{c} = (c_1, \ldots, c_n)^T$. A reasonable scheme would be to minimize (12) iteratively with respect to θ and $\mathbf{c}$. If θ 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, let $\mathbf{g}_\alpha = \mathbf{R}_\alpha \mathbf{c}$, and $\mathbf{G} = [\mathbf{g}_1, \ldots, \mathbf{g}_p]$ be the $n \times p$ matrix with the αth column being $\mathbf{g}_\alpha$. Then (12) 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 θ.

Letting $\mathbf{w}_\theta = (w_{\theta 1}, \ldots, w_{\theta n})^T$, 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

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

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

$$\begin{aligned} & A(\tilde{\theta}) + (\theta - \tilde{\theta})^T G_A(\tilde{\theta}) + \frac{1}{2}(\theta - \tilde{\theta})^T H_A(\tilde{\theta})(\theta - \tilde{\theta}) \\ = & \frac{1}{2}\theta^T H_A(\tilde{\theta})\theta + \theta^T\{G_A(\tilde{\theta}) - H_A(\tilde{\theta})\tilde{\theta}\} + \text{constant}, \end{aligned} \tag{14}$$

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

For fixed λ_0 and M, our algorithm is as follows.

1. Initialization: fix $\theta_\alpha = 1$, $\alpha = 1, \ldots, p$.
2. For a currently given θ, solve for $\mathbf{c}$.
3. For the current $\mathbf{c}$, and $\tilde{\theta}$ being the current iterate of θ, solve for θ in (14).
4. repeat Step 2 and Step 3 until θ converges or a given number of times, whichever comes first.

4.4. Choosing the smoothing parameter

The 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 the 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 (RKL) loss $RKL(p, \hat{p}) = -E_X \log \hat{p}(\mathbf{X})$. If a tuning set is available, we can use the empirical RKL loss on the tuning set to tune the smoothing parameter. When there is no tuning set, we can use five- or ten-fold cross-validation.

In computing the KL loss, we need to evaluate the constant term in (7). This is done through 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 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 $\theta_\alpha = 1$, $\alpha = 1, \ldots, p$. Solve the smoothing spline problem and tune λ_0 according to CV. Set λ_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 used simulated data from a number of sources. All examples considered in this section are for the two-way interaction models.

For the univariate problem the algorithm described in Section 4.2 was directly applied. The COSSO algorithm for the all-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 (referred to as NEW) is compared with the Gaussian kernel density estimation (GKDE) and the penalized likelihood (PL) method. The kernel density estimator used is the simplest kind that involves only one smoothing parameter. It is simple to use and works well for low-dimensional problems, but is not expected 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 was 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 generated an independent test sample $\{\mathbf{x}_k^*,\, k = 1, \ldots, N\}$ from the true density p, and used the empirical KL loss on the test sample to compare the estimators.

Example 5.1. Samples of size $n = 100$ were generated from the univariate density proportional to $(1/3)N(0.3, 0.1^2) + (2/3)N(0.7, 0.1^2)$ truncated to $[0, 1]$. The estimated density functions by our method, the GKDE, and the PL method based on the first sample are shown in Figure 1. For these estimated densities, the empirical KL losses were computed based on an independent test sample of size $N = 1,000$ from the true density, and these were $KL_{NEW} = 0.0119$, $KL_{GKDE} = 0.0220$ and $KL_{PL} = 0.0103$. We ran the simulation $s = 100$ times more, and the empirical KL losses on test samples were computed in each simulation and displayed in the boxplots (Figure 1, bottom right). The boxplots show that the three methods are comparable in this one-dimensional example, that the PL method performs slightly better than the others.

Example 5.2. A 4-dimensional density was constructed by independently combining the univariate density used in Example 5.1 and the two-dimensional density proportional to

$$\frac{1}{2}N((0.3, 0.5), \frac{I}{49}) + \frac{1}{3}N((0.7, 0.7), \frac{I}{49}) + \frac{1}{6}N((0.75, 0.25), \frac{I}{49}) \qquad (15)$$

truncated on $[0, 1]^2$. (X_1, X_2) follows (15), and X_3 and X_4 follow the density in Example 5.1. (X_1, X_2), X_3, and X_4 are independent of each other. Three estimation methods were applied to the $s = 50$ samples of size $n = 600$. The test sample size for the empirical KL was $N = 3,000$.

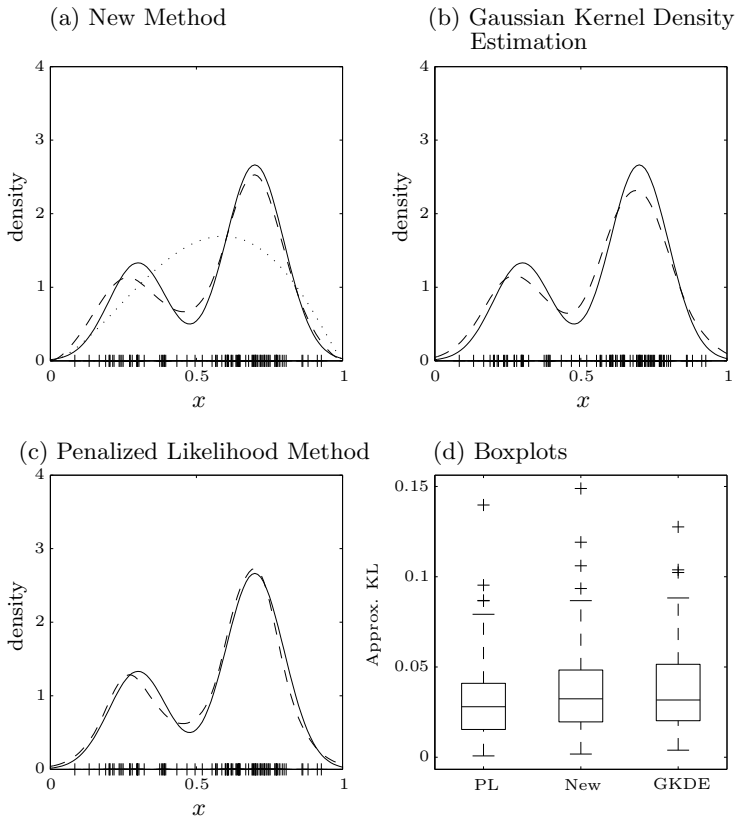


Figure 1. Based on a sample of size $n = 100$ from the true density (solid line) of Example 5.1, the estimated densities (dashed line) by the new method (top left), the Gaussian kernel density estimation (top right), and the penalized likelihood method (bottom left) are plotted. The dotted line in the top left panel indicates the baseline density. Boxplots of the empirical KL losses on test samples 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.

For visual illustration of the joint density estimate, the two-dimensional contour plots of the estimated densities, based on a sample of size $n = 600$ of the random pair (X_1, X_2) in Example 5.2, are displayed along with the true contour plot in Figure 2.

The R function *ssden* for the PL method adopts a sub-basis scheme which uses only a part of the sample as basis functions to reduce the computational load. That is, the minimizer is found in the space spanned by $\{R(\mathbf{x}_i, \cdot),\ i \in I,$ and $\phi_j(\cdot),\ j = 1, \ldots, l\}$ for a random subset I of $\{1, \ldots, n\}$, instead of $\{1, \ldots, n\}$ itself, where $\{\phi_1, \ldots, \phi_l\}$ is a basis of the null space. The cardinality of the set I is denoted by n_B.

For comparison to the PL method, we also brought the sub-basis scheme into our method in this example. To compare the running time between our method and the PL 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 *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.

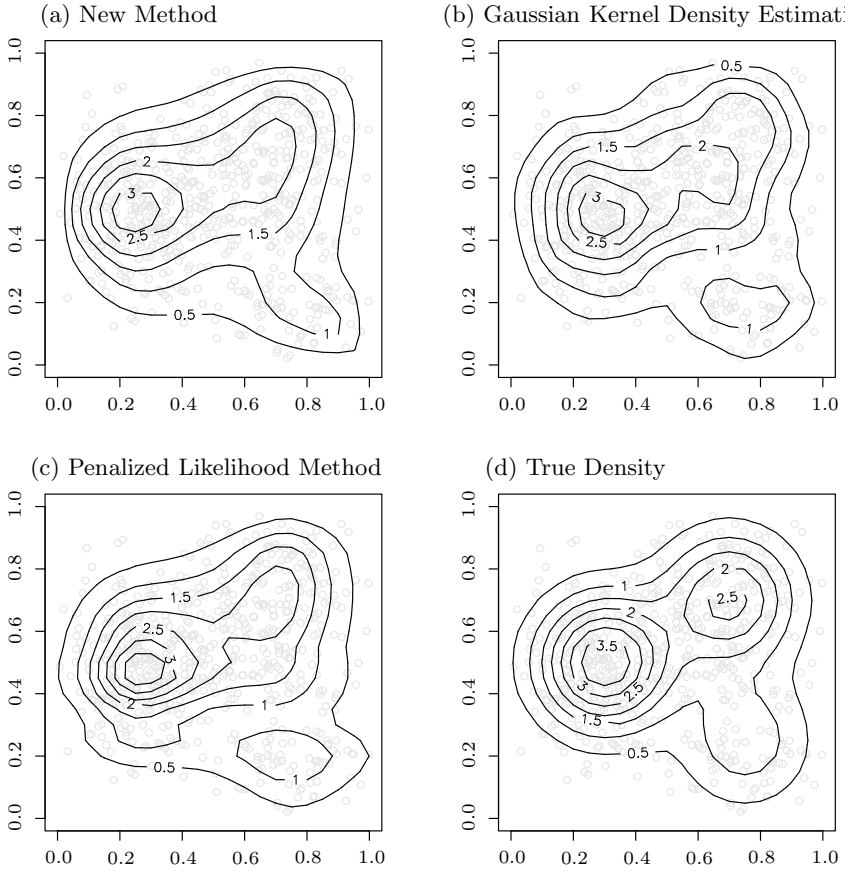


Figure 2. Based on a sample of size $n = 600$ of the pair (X_1, X_2) in Example 5.2, the two-dimensional contour plots of the estimated densities by the new method (top left), the Gaussian kernel density estimation (top right), and the penalized likelihood method (bottom left) are displayed along with the true contour plot (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.

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

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

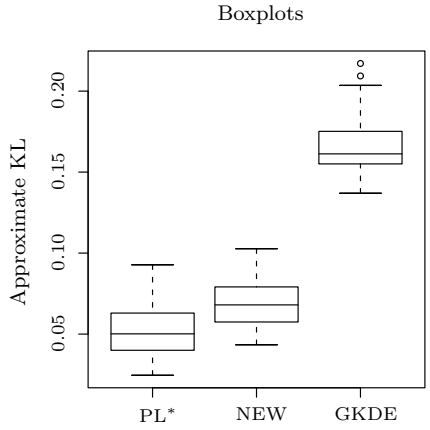


Figure 3. Example 5.2. Boxplots of the empirical KL losses on test samples of estimated densities by our method and the GKDE based on $s = 50$ simulations. The boxplot for the PL method is based on only 47 runs. The PL method failed in the other 3 runs.

To compare the performance of the methods in Example 5.2, we applied our method and the *ssden* function with $n_B = 42$ in $s = 50$ simulations. 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 three cases, where the answers were far from reasonable estimates. The algorithm finished successfully in the remaining cases. Our method had no problems.

Figure 3 shows boxplots of the empirical KL losses on test samples for our method and the GKDE based on $s = 50$ simulations. The boxplot for the PL method based on only 47 cases, where *ssden* was successful, is also displayed in Figure 3, together with the boxplots for our method and the GKDE based on 50 simulations. It is clear that our method performs much better than the GKDE. If we use only the 47 cases where the *ssden* function successfully obtained the estimates to compute the error for the PL method, the PL method performed slightly better than our method. This result is biased in favor of the PL method,

since we effectively deleted the outliers in the performance measure of the PL method.

In higher dimensional problems, it is appropriate to use more basis functions, and the computational cost for integration also increases exponentially. Therefore the PL method is not suitable for high-dimensional problems, its implementation in *gss* can handle at most 4-dimensional problems. Our method is computationally much more practical. For examples with dimension higher than four, comparisons are made only between our method (with the full basis) and GKDE, since the PL estimates cannot be computed.

Example 5.3. Samples of size $n = 600$ were generated from the 5-variate normal density truncated to $[0,1]^5$ with mean $(0.5, 0.5, 0.5, 0.5, 0.5)$ and covariance matrix Σ_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 $X_j \perp\!\!\!\perp X_k \mid$ (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 4, left panel). We repeated the simulation $s = 50$ times, the test sample size was $N = 3,000$.

Example 5.4. Another 5-dimensional density was constructed by independently combining two 2-dimensional densities and the univariate density used in Example 5.1. (X_1, X_2) follows the 2-dimensional density proportional to (15) truncated to $[0,1]^2$, X_5 follows the density of Example 5.1, and (X_3, X_4) follows the 2-dimensional density

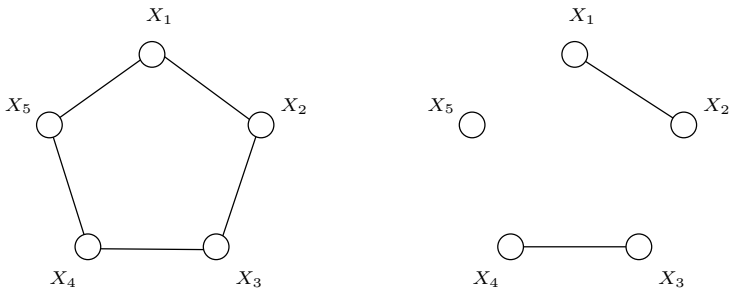
$$\frac{2}{3}\text{Beta}^2(2,4) + \frac{1}{3}\text{Beta}^2(7,4), \tag{16}$$

where Beta^2 represents the 2-dimensional distribution where each variable independently follows the corresponding Beta density. (X_1, X_2), (X_3, X_4), and X_5 are independent of each other. Figure 4 (right panel) shows the corresponding graph. Two estimation methods were applied to the $s = 50$ samples of size $n = 600$, the test sample size was $N = 3,000$.

Example 5.5. A 10-dimensional density was constructed by independently combining $(X_1, \ldots, X_5)$ from Example 5.3, (X_6, X_7, X_8) from the 3-variate normal density truncated to $[0,1]^3$ with mean $(0.5, 0.5, 0.5)$ and covariance matrix Σ_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 (16). Notice that $X_7 \perp\!\!\!\perp X_8 \mid X_6$. We considered only 11 two-way interactions including the 8 interactions present in the true density (the solid edges in Figure 5). The additionally considered interactions are those corresponding to the dashed edges in Figure 5. With sample size $n = 600$, the simulation was repeated $s = 50$ times; the test sample size $N = 5,000$ was used.



(a) Example 5.3. (b) Example 5.4.

Figure 4. Graphs for two 5-D examples. The edges indicate the interaction terms present in the true density.

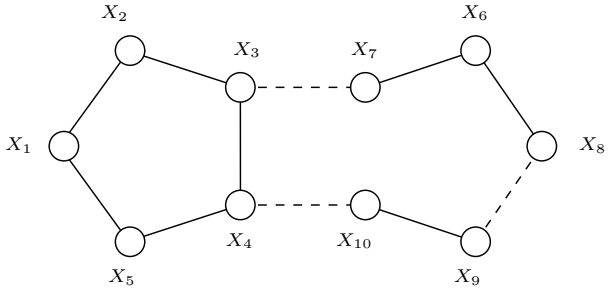


Figure 5. Graph for Example 5.5. The interactions corresponding to the solid edges are those present in the true density. The interactions corresponding to the dashed edges are additionally included in the estimation procedure.

For the Examples 5.3, 5.4 and 5.5, boxplots of the empirical KL losses on test samples for our method and the GKDE based on $s = 50$ simulations are displayed in Figure 6. It is clear that our method performs better than the GKDE, and much better in Example 5.4 and 5.5.

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 θ as zero if it was 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 η_{9T} between X_9 and X_{10} (denoted by T). In the first row of each example, 1 indicates presence of the corresponding term in the true density and 0 indicates 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 η_{34} was missed 10 times, but the interaction η_{23} was missed only once, and all the other correct interactions were never missed. The interactions η_{23} and η_{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. 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.

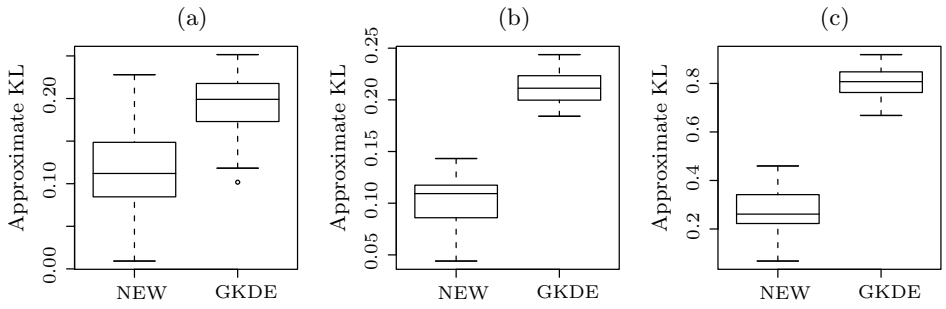


Figure 6. Boxplots of the empirical KL losses on test samples of estimated densities by two methods in (a) Example 5.3 (b) Example 5.4, and (c) Example 5.5 based on $s = 50$ simulations.

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 presence of a term in the true density is indicated by 1 and absence by 0 on top, and the counted frequency is on bottom.

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

6. Example

As a practical application of our method, we consider the following example.

Example 6.6.(NO2) The data originated in a study where air pollution at a road was related to traffic volume and meteorological variables. It was collected by the Norwegian Public Roads Administration. The data set, contributed by Magne Aldrin, consists of a subsample of 500 observations from the original data set with the following variables: (a) hourly values of the logarithm of the concentration of NO2, (b) the logarithm of the number of cars per hour, (c) temperature 2 meter above ground, (d) wind speed, (e) the temperature difference between 25 and 2 meters above ground, (f) wind direction, (g) hour of day, and (h) day number.

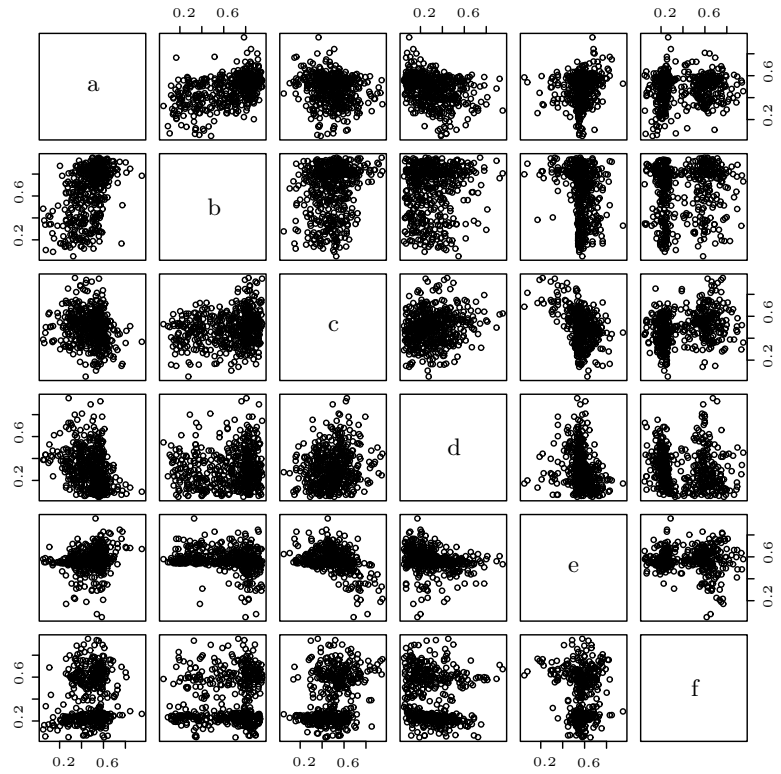
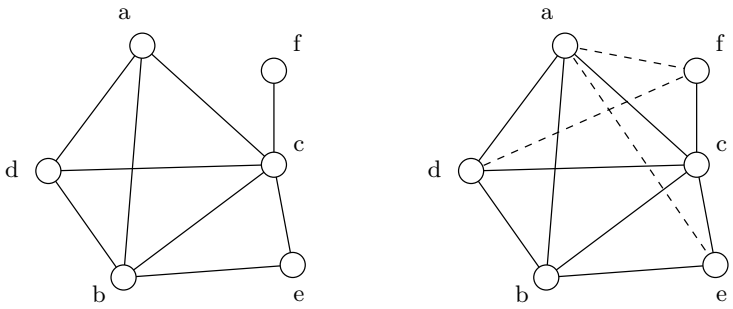


Figure 7. Pairwise scatter plots of NO2 data.

The data is available at the StatLib Datasets Archive at Carnegie Mellon University. The URL is lib.stat.cmu.edu/datasets/. We consider the first six variables (a)−(f), they are all continuous. For the analysis of the data with our method, the variables are scaled so that the values of each dimension fall in $[0, 1]$. The pairwise scatter plots displayed in Figure 7 indicates that it is inappropriate to assume normality. For comparison, however, we took into account

the parametric Gaussian graphical models, and the statistical software package MIM introduced in Edwards (2000) was used for fitting the Gaussian graphical model and for model selection. The default backward stepwise model selection procedure, starting from the saturated model, was used for the MIM.

The selected models by our method and the MIM are shown in Figure 8. The selected model by our method is simpler than that by the MIM. To evaluate the estimation accuracy of the two methods for this example, we randomly separated the data set into a training set (of size 300) and a test set (of size 200), built the model based on the training set, and computed the log-likelihood on the test set. Notice that a larger log-likelihood on the test set indicates better fitting. We repeated the procedure 100 times and computed the mean log-likelihood. The mean log-likelihood for our method was 3.10, and the mean log-likelihood for the MIM was 3.01. Therefore our method provides a simpler model but it fits the data better than the MIM for this example.



(a) Our method. (b) MIM.

Figure 8. The selected models of NO2 data by our method and the MIM. The dashed edges in the right panel indicate the interaction terms selected by the MIM but not by our method.

7. Discussion

An important question in our method is the choice of the smoothing parameters. One possibility is to reserve an independent test set for the tuning. Another possibility is k-fold cross-validation, as used in this paper. It is hoped 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. Another question that needs further investigation is how the estimator behaves for different choices of the baseline density ρ.

Acknowledgement

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

References

Cox, D. D. and O'Sullivan, F. (1990). Asymptotic analysis of penalized likelihood and related estimators. *Ann. Statist.* **18**, 1676-1695.

Edwards, D. (2000). *Introduction to Graphical Modelling*. Springer-Verlag Inc.

Gu, C. (1993). Smoothing spline density estimation: a dimensionless automatic algorithm. *J. Amer. Statist. Assoc.* **88**, 495-504.

Gu, C. (2002). *Smoothing Spline ANOVA Models*. Springer-Verlag Inc.

Gu, C. and Qiu, C. (1993). Smoothing spline density estimation: theory. *Ann. Statist.* **21**, 217–234.

Gu, C. and Wang, J. (2003). Penalized likelihood density estimation: direct cross-validation and scalable approximation. *Statist. Sinica* **13**, 811-826.

Leonard, T. (1978). Density estimation, stochastic processes and prior information. *J. Roy. Statist. Soc. Ser. B* **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.

Silverman, B. W. (1982). On the estimation of a probability density function by the maximum penalized likelihood method. *Ann. Statist.* **10**, 795-810.

Wahba, G. (1990). *Spline Models for Observational Data*. SIAM, Philadelphia.

Whittaker, J. (1990). *Graphical Models in Applied Multivariate Statistics*. John Wiley, New York.

Department of Statistics, University of Wisconsin-Madison, 1300 University Ave., Madison, WI 53706, U.S.A.

E-mail: yjeon@stat.wisc.edu

Department of Statistics, University of Wisconsin-Madison, 1300 University Ave., Madison, WI 53706, U.S.A.

E-mail: yilin@stat.wisc.edu

(Received March 2005; accepted July 2005)