Least Squares Mixture Decomposition Estimation

by

Donggeon Kim

Dissertation submitted to the Faculty of the

Virginia Polytechnic Institute and State University

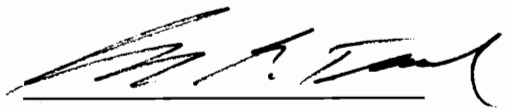
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

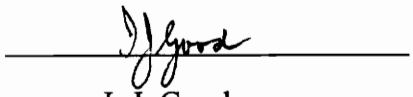
in

Statistics

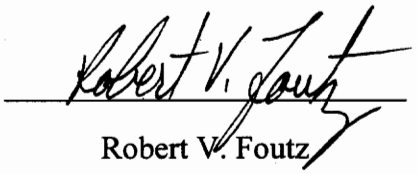
Approved by



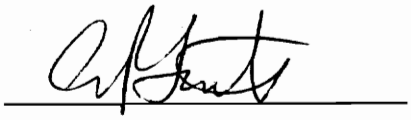
George R. Terrell, Chairman



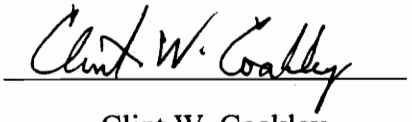
I. J. Good



Robert V. Foutz



Eric P. Smith



Clint W. Coakley

Feburary 13, 1995

Blacksburg, Virginia

LEAST SQUARES MIXTURE DECOMPOSITION ESTIMATION

by

Donggeon Kim

Committee Chairman: George R. Terrell

Statististics

(ABSTRACT)

The Least Squares Mixture Decomposition Estimator (LSMDE) is a new nonparametric density estimation technique developed by modifying the ordinary kernel density estimators. While the ordinary kernel density estimator assumes equal weight ($1/n$) for each data point, LSMDE assigns the optimized weight to each data point via the quadratic programming under the Mean Integrated Squared Error (MISE) criterion. As results, we find out that the optimized weights for a given data set are far different from $1/n$ for a reasonable smoothing parameter and, furthermore, many data points are assigned to zero weights after the optimization. This implies that LSMDE decomposes the underlying density function to a finite mixture distribution of p ($< n$) kernel functions. LSMDE turns out to be more informative, especially in multi-dimensional cases when the visualization of the density function is difficult, than the ordinary kernel density estimator by suggesting the underlying structure of a given data set.

ACKNOWLEDGMENT

I would like to thank my dissertation advisor Professor George R. Terrell for his constant encouragement and numerous valuable suggestions. Without his guidance, I could not have proposed or completed this research.

I am grateful to Professors Robert V. Foutz, Eric P. Smith and Clint W. Coakley for their support. Especially I would like to give special thanks to Professor I. J. Good for his suggestions and advice.

I have also benefited from supportive friends throughout my graduate years, in particular, In-Gyu Kim, Youn-Soo Kang, and Clark Gaylord.

I would like to express my deepest appreciation to my parents, my brother and sisters, and their families. Finally, for their encouragement and patience, I am most grateful to my wife, Young-Seon, and my son, Hyunbin.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Review of density estimation

One of the fundamental characteristics describing the behavior of a random variable X is its probability density function. Specifying the probability density function, we can compute the probability of X over a certain region A,

$$P(X \in A) = \int_A f(x)dx$$

where $f(x)$ is the probability density function of X. Knowledge of the density function provides a way of understanding of underlying random variables. In most cases, the probability density function of a random variable is unknown. Instead, statisticians are given a set of observations. From now on, we suppose we have n independent, identically distributed observations $\{X_i\}$, i=1,...,n from an unknown probability density function, $f(x)$. Let us begin with a brief review of different approaches to density estimation.

1.1.1 Parametric approach

One approach that can be considered as classical statistics is parametric statistical inference. The parametric approach is based on fairly specific assumptions regarding the nature of the underlying distribution. Its form and parameters have to be specified at the

beginning of the analysis. Under these assumptions, the parametric approach is easy to interpret and theoretically well understood.

The observations are assumed to be drawn from one of known parametric families of distributions which is believed to describe an underlying density function fairly well. Then, by estimating and testing the unknown parameters of the parametric family, we have a complete density estimate for the underlying distribution. However, the results from the parametric approach are valid only as long as all assumptions are reasonable. In many cases, the underlying assumptions may be too stringent and its application may be misleading when the assumptions fail. They may not be robust to the presence of slight perturbations in data sets; and, for some cases, we might not even find a suitable parametric family of distributions for our observations.

1.1.2 Nonparametric approach

Another class of methods, the nonparametric approach, does not restrict the possible form of the density function. While the parametric approach concentrates on obtaining the best estimator of unknown parameters, the nonparametric approach focuses on obtaining a good estimate of the entire density function. The main idea behind it is to reconstruct the underlying density function with as few assumptions as possible. Nonparametric methods reduce the need for stringent, and sometimes implausible, assumptions that we have to make in parametric estimation. Due to the lesser need for assumptions, the nonparametric approach is suitable when little information about the form and the family of the true density is available.

The histogram is one of the oldest and the most widely used nonparametric techniques for estimating an unknown density function. The disadvantages of the histogram are its discontinuity and its origin dependency [Silverman (1986)]. There have been various approaches to estimating the unknown density functions to overcome the drawbacks of the histogram. Some examples include extensions of the histogram such as frequency polygons, the average shifted histogram, the nearest neighbor methods, the orthogonal series methods, the maximum penalized likelihood estimators, and the kernel density estimates.

Among these approaches, the kernel estimate is widely applicable and its properties are best understood. Many researchers have analyzed the behavior and the performance of the kernel density estimates and have modified the ordinary kernel density estimates in several directions for better performance. In section 1.2, we will review some basic properties of the kernel density estimate which will be needed in later chapters.

1.1.3 Finite mixture distribution approach

This approach has received increasing attention in the statistical literature recently because of its applicability in many areas. Finite mixture distributions play an important role in modeling heterogeneous data with the focus on applications in the field of cluster analysis. Most standard densities do not have multiple bumps; and the presence of more than one bump in a density is indicative of a mixture [Silverman (1986), p. 137, Good and Gaskins (1980, p. 42)].

Suppose we have a random variable X, which takes values in a sample space Ω, and that its distribution can be represented by a probability density function of the form

$$f(x) = \sum_{i=1}^{k} \pi_i f_i(x) \tag{1.1}$$

where $\pi_i > 0$, $i = 1,..,k$; $\sum_{i=1}^{k} \pi_i = 1$, and $f_i(x)$ is a probability density function for $i = 1,..,k$.

In such a case, X is defined to have a finite mixture distribution with finite mixture density function $f(x)$ in (1.1). The parameters $\{\pi_i\}$, $i = 1,..,k$ are called the mixing weights and $f_i(x)$, $i = 1,..,k$ the component densities of the mixture [Titterington, Smith and Makov (1985) p. 1]. The component densities are usually assumed to be from the same distribution with different parameters for all $i = 1,..,k$. The goal of the finite mixture distribution is the estimation of the parameters of the mixture distribution such as mixing weights or the number of mixing component densities. Finite mixture distribution techniques, however, can also be used to approximate an unknown density function and are closely related to the kernel density estimates [Titterington, Smith, and Makov (1985) p. 28, Everitt and Hand (1981) Sec 5.3]. This will be illustrated while we explain the motivation of our new density estimator.

1.2 Kernel density estimator

1.2.1 Definition of kernel density estimators

The kernel density estimate of $f(x)$ is defined by

$$\hat{f}(y) = \frac{1}{nh}\sum_{i=1}^{n} K(\frac{y - x_i}{h}) = \frac{1}{n}\sum_{i=1}^{n} K_h(y - x_i) \tag{1.2}$$

where $K_h(t) = \frac{1}{h}K(\frac{t}{h})$, known as the kernel function and the smoothing parameter h. In general, kernel functions are assumed to be both positive, symmetric about zero, and have integral 1. For these reasons, symmetric probability density functions are one of the popular choices for kernel functions. This assures the kernel density estimate to be a density function itself. We may relax these conditions, especially the positivity condition of kernel functions. If the kernel estimate is allowed to be negative for part of its range, significant improvement can be achieved [Scott (1992) p. 133]. Since we want to estimate underlying density functions using a finite mixture of probability density functions, we restrict kernel functions to be probability density functions.

The kernel density estimator is indeed a density function and can be interpreted as a mixture density of n equally weighted kernels each centered at a data point [Titterington, Smith, and Makov (1985) p. 28]. Provided that $K(u)$ is a probability density function, the kernel density estimate has n mixing components with equal mixing weights of $1/n$.

The kernel density estimator is determined by the kernel function $K(u)$, and the smoothing parameter, h. The kernel function determines the shape of the bumps while the smoothing parameter h determines their width. The property of smoothness and

differentiability of kernel functions is inherited by $\hat{f}(y)$. Therefore, if $K(u)$ is n times continuously differentiable, $\hat{f}(y)$ is n times continuously differentiable.

As $h \to 0$, $\hat{f}(y)$ gets rough. As $h \to \infty$, $\hat{f}(y)$ gets smoother. There is a well-known trade-off relationship between the bias and the variance of the resulting density estimates. Since the choice of certain kernel function is not so critical to $\hat{f}(y)$ in the asymptotic sense, we usually select the kernel based on the degree of differentiability or computational efficiency [Silverman (1987) p. 43]. On the other hand, the appropriate choice of the smoothing parameter h has been of crucial importance. Most current research has concentrated on how to determine the desirable smoothing parameter in systematic ways according to certain criteria.

1.2.2 Statistics of kernel density estimators

This section will be a review of some criteria that have been developed to measure the discrepancy of the density estimator $\hat{f}(y)$. Let us begin with pointwise analysis of the kernel estimates. The mean square error of $\hat{f}$ at $x = y$ is defined as

$$MSE(\hat{f}(y)) = E_y[\hat{f}(y) - f(y)]^2 = Bias^2[\hat{f}(y)] + Var[\hat{f}(y)]. \tag{1.3}$$

For a symmetric non-negative kernel, the bias and the variance of $\hat{f}_K(y)$ are given as

$$Bias(\hat{f}_K(y)) = \frac{1}{2}\sigma_K^2 h^2 f''(y) + O(h^4),$$

$$Var(\hat{f}_K(y)) = \frac{1}{nh} f(y) R(K) - \frac{f(y)^2}{n} + O\left(\frac{h}{n}\right) \tag{1.4}$$

where $R(f) = \int f^2(y)dy$ and $\sigma_K^2 = \int y^2 K(y)dy$ [Scott (1992) p. 130].

$\hat{f}_K(y)$ is asymptotically unbiased as $h \to 0$, and is consistent in MSE if $h \to 0$, and $nh \to \infty$. We observe a trade-off problem in the bias and the variance from (1.4). The variance is proportional to $1/(nh)$, suggesting that the larger h becomes, the smaller the variance. However, choosing h larger makes the bias of $\hat{f}_K(y)$ inflated. Therefore, the smoothing parameter h plays a role of compromising the bias and the variance of the resulting estimate.

Since we are most interested in the behavior of the estimate over the whole possible range, we need global criteria so that we can take into account the entire density surface in order to compare different density estimators. Some of them are based on norms, the L_∞ norm ,($\sup_y |\hat{f}(y) - f(y)|$), the L_1 norm ($\int |\hat{f}(y) - f(y)| dy$), or the L_2 norm ($\int [f(y) - \hat{f}(y)]^2 dy$). Because of the tractability of the L_2 norm, *integrated squared error* (ISE) = $\int [f(y) - \hat{f}(y)]^2 dy$ is the most widely used criterion for the global accuracy of $\hat{f}(y)$. For most cases, it is sufficient to examine the average of ISE or the *mean integrated squared error* (MISE) defined by

$$MISE(\hat{f}) = E\{\int [\hat{f}(y) - f(y)]^2 dy\} \tag{1.5}$$

It is known that the asymptotic MISE (AMISE) of the kernel density estimator with a non-negative kernel density estimator is

$$AMISE = \frac{R(K)}{nh} + \frac{1}{4}\sigma_K^4 h^4 R(f'') \tag{1.6}$$

where $R(f) = \int f^2(y)dy$ and $\sigma_K^2 = \int y^2 K(y)dy$ [Scott (1992) p. 131]. The optimal smoothing parameter h^*, which minimizes the AMISE is

$$h^* = \left[\frac{R(K)}{\sigma_K^4 R(f'')}\right]^{1/5} n^{-1/5}, \tag{1.7}$$

with

$$AMISE^* = \frac{5}{4}\left[\sigma_K R(K)\right]^{4/5} R(f'')^{1/5} n^{-4/5} \tag{1.8}$$

[Scott (1992), (6.16)].

In practice, the underlying density function is unknown and hence we are unable to obtain the optimal smoothing parameter h^*. Assuming the true density function is normal, we often use a normal reference rule for smoothing parameter using a normal kernel with $h = 1.06\hat{\sigma}\, n^{1/5}$ [Scott (1992) p. 131].

There have been two different approaches to improving the kernel estimation, one in the choice of a kernel function, and the other in the choice of a smoothing parameter. Even though it is well known that the choice of a specific kernel function affects a density estimate in a minor way, significant improvement in the rate of convergence of the estimate can be achieved by so called higher-order kernels if we allow $K(u)$ to take on negative values. It is known that, if $K(u)$ is chosen so that its first $p-1$ moments vanish (known as a kernel of order p), and if $f(x)$ has p continuous derivatives , then the rate of convergence of kernel estimate is $O(n^{-2p/(2p+1)})$ [Scott (1992) p.133]. This improvement is done by reducing the contribution of the bias to the MISE. An alternative method of reducing bias is proposed by Terrell and Scott (1980).

A lot of research in density estimation focused on data-driven selection methods of the optimal bandwidth. Cross-validation (leave-one-out method) is one of popular methods to choose the optimal bandwidth. Cross-validation methods have several different versions according to the choice of criterion.

1. Maximum likelihood cross-validation maximizing $\prod_{i=1}^{n} \hat{f}_{-i}(x_i; h)$

where $\hat{f}_{-i}(x_i;h)$ is the leave-one-out estimate.

2. Least squares cross-validation minimizing an unbiased estimate of MISE [Rudemo (1982), Bowman (1984)].
3. Biased cross-validation based on a slightly biased but less variable estimator of MISE [Scott and Terrell (1987)].

A split-sample procedure was proposed for the choice of the smoothing parameter in the maximum penalized likelihood method by Good and Holtzman (1989).

Finally, we comment on two different ideas based on the maximum likelihood methods and unequal weights on the kernel estimate. Geman and Hwang (1982) suggest the maximization of likelihood function in the sequence of collections of densities called a sieve. They give an estimate of the form, for the normal convolution sieve,

$$\hat{f}_S(y) = \sum_{i=1}^{n} p_i \frac{1}{h} K\left(\frac{y - y_i}{h}\right) \tag{1.9}$$

for some probability vector $(p_1, p_2, \ldots, p_n)$ and some real numbers $(y_1, y_2, \ldots, y_n)$ all strictly contained in the range of data set. They found out that the ordinary kernel estimate is in the sieve, but not among the optimal solutions [Devroye and Györfi (1985)]. Also Scott (1976) proposed the optimal weights, $\{a_i\}_{i=1}^{n}$ that solve the following constrained optimization problem:

$$\begin{aligned} \text{maximize} \quad & \prod_{i=1}^{n} \hat{f}(y) = \prod_{i=1}^{n} \sum_{i=1}^{n} a_i \frac{1}{h} K\left(\frac{y - x_i}{h}\right) \\ \text{subject to} \quad & a_i \geq 0, \text{ for all } i = 1, \ldots, n \\ & \sum_{i=1}^{n} a_i = 1 \quad . \end{aligned} \tag{1.10}$$

He established the existence and uniqueness of the solution of optimization problem (1.10) and reported that most of the optimized weights are set equal to zero.

Chapter 2

The Least Squares Mixture Decomposition Estimator

In this chapter, we present the least squares mixture decomposition estimator (LSMDE) as a new density estimator, which is a generalization of ordinary kernel density estimators with the connection to the theory of finite mixture distributions. From the link between these two methods, the LSMDE provides more interpretable explanation about underlying distributions than the kernel estimator and can be easily applied to the mixture problem or discriminant analysis.

2.1 Definition of the LSMDE

2.1.1 Motivation

While most researches on the kernel density estimate have been focusing on the determination of a desirable smoothing parameter h, researchers have always assumed that each data point receives the equal weight, $1/n$. By definition, the kernel estimator is the arithmetic mean of n independent and identically distributed random variables,

$$K_h(y - X_i) = \frac{1}{h} K\left(\frac{y - X_i}{h}\right).$$

By inspecting the definition, we can interpret the kernel density estimates for a random sample $\{X_i\}$, i=1,...,n, as equally weighted mixture densities. We can readily connect the kernel estimates with the theory of mixture distributions by replacing the weights from $1/n$ with general weights $\{a_i\}$, i=1,...,n. Therefore, (1.2) can be rewritten as

$$\hat{f}(y) = \frac{1}{n}\sum_{i=1}^{n} K_h(y - x_i) = \sum_{i=1}^{n} a_i K_h(y - x_i) \tag{2.1}$$

where $a_i = \frac{1}{n}$, for all i=1,...,n.

The question is whether, for fixed h, and under some criterion, we can have a better estimate in the form of

$$\hat{f}(y) = \frac{1}{h}\sum_{i=1}^{n} a_i K(\frac{y - x_i}{h}) = \sum_{i=1}^{n} a_i K_h(y - x_i) \tag{2.2}$$

where

$$\sum_{i=1}^{n} a_i = 1 \text{ and } a_i \geq 0 \quad \text{for all } i \tag{2.3}$$

The two constraints (2.3) on the weights guarantee that the new estimator becomes a probability density function itself. The first constraint assures that the estimate integrates to 1. The second constraint guarantees that the estimate is always positive over $(-\infty, \infty)$. We restrict ourselves to the one with the weight condition (2.3), since the relaxation of the weight condition no longer permits the interpretation of new estimator as a mixture estimator. As we have mentioned, Scott (1976) and Geman and Hwang (1980) propose the possibility of unequal weights of the kernel estimate.

2.1.2 Objective function

We wish to obtain the weights $\{a_i\}$, i=1,..,n under some criterion that will assure a good density estimate. We choose the ISE as our discrepancy criterion and we want to have density function estimates of the form (2.2) subject to (2.3), which minimizes an objective function reflecting ISE based on the observations we have.

Given any density estimator $\hat{f}(y)$ of a density $f(y)$, the ISE can be written as

$$ISE[\hat{f}(y)] = \int [f(y) - \hat{f}(y)]^2 dy = \int f^2 - 2\int f\,\hat{f} + \int \hat{f}^2$$
$$= \int f^2 + Q[\hat{f}(y)] \qquad (2.5)$$

As the first term, $\int f^2$ in (2.5) is unknown but fixed and does not depend on $\hat{f}(y)$, minimization of the ISE depends on the last two terms or $Q[\hat{f}(y)]$. Minimization of $Q[\hat{f}(y)]$, however, still depends on the unknown density $f(y)$. We replace the middle term $\int f\,\hat{f} = E[\hat{f}]$ by its unbiased estimator $\hat{E}[\hat{f}] = \frac{1}{n}\sum_{i=1}^{n} \hat{f}(X_i)$ so that we can have an objective function directly accessible from the observations. The objective function we wish to minimize becomes an estimator of $Q[\hat{f}(y)]$,

$$\hat{Q}[\hat{f}(y)] = -\frac{2}{n}\sum_{i=1}^{n} \hat{f}(X_i) + \int \hat{f}^2(y)dy \qquad (2.6)$$

Note that $\int \hat{f}^2(y)dy$ depends only on the data, *K(u)*, and the smoothing parameter *h*, since

$$\int \hat{f}^2(y)dy = \int [\sum_{i=1}^{n}\sum_{j=1}^{n} a_i a_j K_h(y - x_i) K_h(y - x_j)]\, dy$$

$$= \sum_{i=1}^{n}\sum_{j=1}^{n} a_i a_j \int K_h(y-x_i)K_h(y-x_j)dy$$
$$= \sum_{i=1}^{n}\sum_{j=1}^{n} a_i a_j K_h^*(x_j-x_i) \qquad (2.7)$$

where $K_h^*(x_j - x_i)$ is the convolution of $K_h(y-x_i)$ and $K_h(y-x_j)$.

As a result, (2.6) depends only on the random sample and the kernel function. Hence it can be explicitly calculated and minimized subject to the constraint (2.3) from the data set. $\hat{Q}[\hat{f}(y)]$ is, in fact, a modification of the least squares cross validation function used for automatic smoothing parameter selection, developed independently by Rudemo (1982) and Bowman (1984).

2.1.3 Definition and conjectures

We define the least squares mixture decomposition estimator (LSMDE) of $f(y)$ according to the objective function (2.6) and the constraint set (2.3).

Definition : LSMDE of $f(y)$

$\hat{f}(y)$ is defined as the least squares mixture decomposition estimator of $f(y)$ if given an i.i.d random sample of size *n*, $\{X_i\}$ i=1,..,*n* , the kernel function $K(u)$, and the fixed smoothing parameter *h* ,

$$\hat{f}(y) = \sum_{i=1}^{n} a_i K_h(y-x_i) \qquad (2.8)$$

where $\{a_i\}$, i=1,..,*n* minimize the objective function,

$$\hat{Q}[\hat{f}(y)] = -\frac{2}{n}\sum_{i=1}^{n}\hat{f}(X_i) + \int \hat{f}^2(y)dy$$

subject to $\sum_{i=1}^{n} a_i = 1$ and $a_i \geq 0$ for all i. (2.9)

The behavior of $\hat{f}(y)$ depends on which a_i's are strictly positive after the minimization of $\hat{Q}[\hat{f}(y)]$ subject to the constraint. If all a_i's are positive and close to $1/n$, the LSMDE must be similar to the ordinary kernel density estimate. If that is not the case, the set of nonzero weights controls the characteristics of the LSMDE. Suppose $\{\hat{\pi}_j\}$, j=1,..,k is the set of positive weights among $\{a_i\}$, $i = 1,...n$ after optimization. Then the LSMDE (2.8) reduces to

$$\hat{f}(y) = \sum_{j=1}^{k}\hat{\pi}_j K_h(y - x_j) \tag{2.10}$$

where $\{\hat{\pi}_j\}$, j=1,..,k is the set of positive weights after optimization.

Note that (2.10) can be interpreted as the density estimate of finite mixture distribution with $\{\hat{\pi}_j\}$, the mixing weights estimates, k, the estimated number of mixing components and $K_h(y - x_i)$, component density [Titterington, Smith and Makov (1985) p. 1]. In other words, the LSMDE becomes an estimate of the underlying probability density function as a finite mixture density, if the optimized weights are not $1/n$ for each observation.

2.2 Estimation of the LSMDE

In the section 2.1, we discussed the criterion for choosing the optimized weights under an ISE-based objective function. Here we present how we actually compute the weights $\{a_i\}$, by the minimization of the objective function, $\hat{Q}[\hat{f}(y)]$ subject to the constraints. This optimization turns out to be a typical quadratic programming problem from optimization theory.

2.2.1 Quadratic programming for the LSMDE

The optimized weight, $\{a_i\}$, i=1,..,n in the LSMDE (2.8) can be obtained by solving the following constrained optimization problem.

$$\begin{aligned} &\min \ \hat{Q}(\hat{f};\underline{a}) = -\frac{2}{n}\sum_{i=1}^{n}\hat{f}(x_i) + \int \hat{f}^2(y)dy \\ &\text{s.t.} \quad \sum_{i=1}^{n} a_i = 1 \qquad \text{and} \qquad a_i \geq 0 \quad \text{for all } i\,. \end{aligned} \tag{2.11}$$

This optimization is categorized as a quadratic programming problem in operations research. Let us translate (2.11) into matrix notation as follows to demonstrate the quadratic programming setup.

The first part of the objective function (linear term) becomes

$$-\frac{2}{n}\sum_{i=1}^{n}\hat{f}(x_i) = -\frac{2}{n}\sum_{i=1}^{n}\sum_{j=1}^{n} a_i K_h(x_i - x_j) = \underline{a}^T \underline{d} \tag{2.12}$$

where $\underline{d} = -\frac{2}{n}\left(\sum_{j=1}^{n} K_h(x_1 - x_j), \quad \sum_{j=1}^{n} K_h(x_2 - x_j), \quad \cdots \quad , \quad \sum_{j=1}^{n} K_h(x_n - x_j) \right)^T$. (2.13)

The second part of the objective function (quadratic term) becomes

$$\int \hat{f}^2(y)dy = \int \sum_{i=1}^{n}\sum_{j=1}^{n} a_i a_j K_h(y - x_i)K_h(y - x_j)dy = \frac{1}{2}\underline{a}^T C \underline{a} \tag{2.14}$$

where $C = 2\begin{bmatrix} K_h^*(0) & K_h^*(x_1 - x_2) & \cdots & K_h^*(x_1 - x_n) \\ K_h^*(x_2 - x_1) & K_h^*(0) & \cdots & K_h^*(x_2 - x_n) \\ \vdots & \vdots & \ddots & \vdots \\ K_h^*(x_n - x_1) & K_h^*(x_n - x_2) & \cdots & K_h^*(0) \end{bmatrix}$, (2.15)

and $K_h^*(x_j - x_i) = \int K_h(y - x_i)K_h(y - x_j)dy = \int K_h(z)K_h(z - (x_j - x_i))dz$ (2.16)

The linear equality constraint is $\underline{1}^T \underline{a} = 1$ and the linear inequality (the positivity condition of the weights) can be rewritten as $I_n \underline{a} \geq \underline{0}$ in matrix notation.

Therefore, the constrained optimization problem (2.11) can be expressed as a typical quadratic programming problem [Fletcher (1987)]:

$$\begin{aligned} \min_{\underline{a}} \quad & \hat{Q}[\hat{f};\underline{a}] = \underline{a}^T \underline{d} + \frac{1}{2}\underline{a}^T C \underline{a} \\ \text{s.t.} \quad & \underline{1}^T \underline{a} = 1 \\ & I_n \underline{a} \geq \underline{0} \end{aligned} \tag{2.17}$$

The objective function is quadratic and the constraint set is linear. While the minimization of $\hat{Q}[\hat{f}(y)]$ without the constraint set leads to the systems of linear equations, $C\underline{a} = \underline{d}$ which can be easily solved for $\underline{a}$, linearly constrained quadratic programming problems are much more difficult to deal with than the unconstrained case.

Algorithms developed for the constrained least squares linear regression can be applied to solve (2.17) provided that C is positive definite. This relationship between the quadratic programming and the constrained least square estimation is summarized in the Appendix A.2. Hager, Jorst and Padalos (1993) gives a survey of recent developments relating to quadratic programming algorithms. Basic properties of the quadratic programming will be presented in the next section while we prove the properties of the LSMDE.

2.2.2. Existence and uniqueness of the LSMDE

We establish the existence of the solution for (2.17) from optimization theory as follows. The quadratic programming problem belongs to convex programming provided that the Hessian matrix of the objective function, C, is positive semi-definite. Since the convex quadratic programming problem has a global solution if C is positive semi-definite and the solution is unique if C is positive definite [Fletcher (1987) p. 229], we need to prove that C is positive semi-definite for the existence and that C is positive definite for the uniqueness of the LSMDE for a given data set. We will show that C is positive definite even in the presence of duplicate data points.

First, we examine how to deal with duplicates in the data set before we show that C is at least positive semi-definite. Suppose there are m distinct data points among n data points such that $(X_1,\ldots,X_1|X_2,\ldots,X_2|\cdots\cdots|X_m,\ldots,X_m)$, where r_j denotes the number of repetitions of X_j, so that $n=\sum_{j=1}^{m} r_j$. Then the Hessian matrix becomes

$$\frac{1}{2}C=\begin{bmatrix} K_h^*(0)\,J_{(r_1\times r_1)} & K_h^*(X_1-X_2)\,J_{(r_1\times r_2)} & \cdots & K_h^*(X_1-X_m)\,J_{(r_1\times r_m)} \\ K_h^*(X_2-X_1)\,J_{(r_2\times r_1)} & K_h^*(0)\,J_{(r_2\times r_2)} & \cdots & K_h^*(X_2-X_m)\,J_{(r_2\times r_m)} \\ \vdots & \vdots & \ddots & \vdots \\ K_h^*(X_m-X_1)\,J_{(r_m\times r_1)} & K_h^*(X_m-X_2)\,J_{(r_m\times r_2)} & \cdots & K_h^*(0)\,J_{(r_m\times r_m)} \end{bmatrix} \tag{2.18}$$

where $J_{(r_j\times r_k)}$ is a matrix of ones of size $(r_j\times r_k)$. The unknown vector, $\underline{a}$ can be partitioned to $(a_{11},\ldots,a_{1r_1}|a_{21},\ldots,a_{2r_2}|\cdots|a_{m1},\ldots,a_{mr_m})^T=(\underline{a}_1,\ldots,\underline{a}_m)^T$ according to the duplicate structure. The quadratic term becomes

$$\begin{aligned} &\frac{1}{2}\underline{a}^T C\underline{a} \\ &=(\underline{a}_1,\ldots,\underline{a}_m)^T\begin{bmatrix} K_h^*(0)\,J_{(r_1\times r_1)} & K_h^*(X_1-X_2)\,J_{(r_1\times r_2)} & \cdots & K_h^*(X_1-X_m)\,J_{(r_1\times r_m)} \\ K_h^*(X_2-X_1)\,J_{(r_2\times r_1)} & K_h^*(0)\,J_{(r_2\times r_2)} & \cdots & K_h^*(X_2-X_m)\,J_{(r_2\times r_m)} \\ \vdots & \vdots & \ddots & \vdots \\ K_h^*(X_m-X_1)\,J_{(r_m\times r_1)} & K_h^*(X_m-X_2)\,J_{(r_m\times r_2)} & \cdots & K_h^*(0)\,J_{(r_m\times r_m)} \end{bmatrix}\begin{pmatrix}\underline{a}_1\\ \vdots\\ \underline{a}_m\end{pmatrix} \\ &=\frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m}\underline{a}_i^T K_h^*(X_i-X_j)J_{(r_i\times r_j)}\underline{a}_j \\ &=\frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m}K_h^*(X_i-X_j)b_i b_j \\ &=\frac{1}{2}\underline{b}^T C^*\underline{b} \end{aligned} \tag{2.19}$$

where $\underline{b}=\left\{b_i=\sum_{k=1}^{r_i}a_{ik}\right\}_{i=1}^{m}$ (m x 1) vector,

$$C^*=2\begin{bmatrix} K_h^*(0) & K_h^*(X_1-X_2) & \cdots & K_h^*(X_1-X_m) \\ K_h^*(X_2-X_1) & K_h^*(0) & \cdots & K_h^*(X_2-X_m) \\ \vdots & \vdots & \ddots & \vdots \\ K_h^*(X_m-X_1) & K_h^*(X_m-X_2) & \cdots & K_h^*(0) \end{bmatrix}. \tag{2.20}$$

The Linear term becomes

$$\begin{aligned} & \underline{a}^T \underline{d} \\ & = (\underline{a}_1, \ldots, \underline{a}_m)^T \begin{pmatrix} d_1 \underline{1}_{r_1} \\ \vdots \\ d_m \underline{1}_{r_m} \end{pmatrix} \\ & = \underline{b}^T \underline{d}^* \end{aligned} \tag{2.21}$$

where $\underline{d}^* = (d_1, \cdots, d_m)^T$.

Note that the modified unknown vector $\underline{b}$ satisfies the constraints exactly the same as $\underline{a}$ does. Therefore the original problem of size n reduces to quadratic programming problem of size m as follows

$$\begin{aligned} \text{minimize} \quad & \hat{Q}[\hat{f}; \underline{b}] = \underline{b}^T \underline{d}^* + \frac{1}{2} \underline{b}^T C^* \underline{b} \\ \text{s.t.} \quad & \underline{1}^T \underline{b} = 1, \; I_m \underline{b} \geq \underline{0} . \end{aligned} \tag{2.22}$$

X_j will receive the optimized weight, b_j which is $\sum_{k=1}^{r_j} a_{jk}$. Therefore the LSMDE with duplicates can always be converted to the one without the duplicates and we assume that there is no duplicate in the data set from now on.

We establish the existence of the LSMDE.

Proposition 2.1 : The Existence of LSMDE

The LSMDE (2.8) under (2.9) always exists.

Proof: It is sufficient to prove C is positive semi-definite.

The (i, j)-th element of $\frac{1}{2}C$, is defined by

$$K_h^*(x_j - x_i) = \int K_h(y - x_i)K_h(y - x_j)dy = \int K_h(z)K_h(z - (x_j - x_i))dz.$$

Note that

$$\begin{aligned}\frac{1}{2}C &= \{ \int K_h(y - x_i)K_h(y - x_j)dy \}_{i=1}^{n}{}_{j=1}^{n} \\ &= \int \{ K_h(y - x_i)K_h(y - x_j) \}_{i=1}^{n}{}_{j=1}^{n} dy \\ &= \int \underline{k}\underline{k}^T dy\end{aligned}$$

where $\underline{k} = \{K_h(y - x_i)\}_{i=1}^n$, ($n$ x 1) column vector.

$\underline{k}\underline{k}^T$ is positive semi-definite by definition, since for all nonzero vector $\underline{v}$ in R^n,

$$\underline{v}^T \underline{k}\underline{k}^T \underline{v} = (\underline{v}^T \underline{k})(\underline{v}^T \underline{k})^T = (\underline{v}^T \underline{k})^2 \geq 0.$$

Therefore, $\frac{1}{2}C = \int \underline{k}\underline{k}^T dy$ is also positive semi-definite.■

Before we prove the uniqueness of the LSMDE, let us review the definition of positive definite functions and Bochner's theorem on characteristic functions which will be used for the uniqueness.

Definition : Positive semi-definite function [Lukacs (1960)]

A complex-valued function $\varphi(t)$ of the real variable t is said to be non-negative definite (or positive semi-definite) for $-\infty < t < \infty$, if the following two conditions are satisfied:

(a) $\varphi(t)$ is continuous;

(b) for any positive integer n and any real $t_1,\ldots t_n$ and any complex$\xi_1,\ldots,\xi_n$,

$$S = \sum_{i=1}^{n}\sum_{j=1}^{n}\varphi(t_i - t_j)\xi_i\bar{\xi}_j \text{ is real and non-negative}$$

where $\bar{\xi}_j$ denotes the complex conjugate of ξ_j. If S is strictly positive whenever $(\xi_1,\ldots,\xi_n)$ is non-zero vector , $\varphi(t)$ is said to be positive definite [Karlin (1967) p. 186].

Properties: The positive semi-definite function $\varphi(t)$ has following properties;

(a) $\varphi(0)$ is real and $\varphi(0) \geq 0$,

(b) $\varphi(-t) = \overline{\varphi(t)}$,

(c) $|\varphi(t)| \leq \varphi(0)$.

Theorem : Bochner's Theorem [Lukacs (1960)]

A complex-valued function $\varphi(t)$ of a real variable t is a characteristic function if and only if (a) $\varphi(t)$ is positive semi-definite, (b) $\varphi(0) = 1$.

Proposition 2.2: The Uniqueness of LSMDE

The LSMDE (2.8) under (2.9) is unique provided that the kernel function is a continuous symmetric unimodal probability density function.

Proof: It is sufficient to show that $K_h^*(\Delta_{ij} = x_i - x_j)$ is a positive definite function for all $x_i \neq x_j$, $i \neq j$. We point out that $K_h^*(\Delta_{ij})$ is real-valued, continuous, at least positive semi-definite (see Proposition 2.1), and symmetric unimodal (about 0) function, since $K_h^*(\Delta_{ij})$ is the convolution of two symmetric unimodal distributions [Dhamadhikari and Joag-dev, (1988), Proposition (vi) in p. 2 or Theorem 1.6].

Consider $\phi(\Delta_{ij}) = K_h^*(\Delta_{ij}) / K_h^*(0)$ so that $\phi(0) = 1$. Note that $\phi(\Delta_{ij})$ satisfies the sufficient condition for a characteristic function of some random variable Y by Bochner's theorem and $\phi(\Delta_{ij})$ is always non-negative and real-valued. By the definition of real-valued characteristic function, we have

$$\phi(\Delta_{ij}) = E[\cos(\Delta_{ij} Y)]$$

where Y is a random variable with $\phi(\Delta_{ij})$ as its characteristic function.

It is sufficient to prove $S = \sum_{i=1}^{n}\sum_{j=1}^{n} a_i a_j \phi(x_i - x_j) > 0$ for any real $x_1, x_2, \ldots, x_n$, and $a_1, \ldots, a_n$ provided $(a_1, \ldots, a_n)^T \neq \underline{0}$.

Since $\phi(\Delta_{ij})$ is at least positive semi-definite, we have inequality

$$S = \sum_{i=1}^{n}\sum_{j=1}^{n} a_i a_j \phi(x_i - x_j)$$

$$= \sum_{i=1}^{n}\sum_{j=1}^{n} a_i a_j E[\cos((x_i - x_j)Y)]$$

$$= \sum_{i=1}^{n}\sum_{j=1}^{n} a_i a_j E[\cos(x_i Y)\cos(x_j Y) + \sin(x_i Y)\sin(x_j Y)] \text{ by trigonometric identity}$$

$$= E[\sum_{i=1}^{n} a_i \cos(x_i Y)\sum_{i=1}^{n} a_j \cos(x_j Y)] + E[\sum_{i=1}^{n} a_i \sin(x_i Y)\sum_{i=1}^{n} a_j \sin(x_j Y)]$$

$$= E[\sum_{i=1}^{n} a_i \cos(x_i Y)]^2 + E[\sum_{i=1}^{n} a_i \sin(x_i Y)]^2 \geq 0.$$

Suppose $S = 0$. Then $\sum_{i=1}^{n} a_i \cos(x_i Y) = \underline{a}^T \underline{c} = 0$ and $\sum_{i=1}^{n} a_i \sin(x_i Y) = \underline{a}^T \underline{s} = 0$ almost everywhere for any real $x_i \neq x_j$, $i \neq j$, and $(a_1, \ldots, a_n)^T \neq \underline{0}$. Since only $\underline{0}$ is orthogonal to

every vector in R^n, $\underline{c}$ and $\underline{s}$ should be $\underline{0}$ in order for S to be zero. Let $A_i = \{Y \mid \cos(x_i Y) = 0\}$ so that $A = \bigcap_{i=1}^{n} A_i$ becomes the solution set of $\underline{c} = \underline{0}$. A_i has a countable number of solutions, since $\cos(x_i Y)$ is a periodic function with period $2\pi / x_i$. Therefore A also has a countable number of solutions. This implies $\mu(A) = 0$ where μ is the probability measure for Y. In similar way, $\mu(B) = 0$ where $B = \bigcap_{i=1}^{n} B_i = \bigcap_{i=1}^{n} \{Y \mid \sin(x_i Y) = 0\}$. Hence the set where S=0 has measure zero and this implies that S is strictly positive almost everywhere.■

Remark 1 : Positive definiteness of normal kernels.

When the kernel function is a standard normal density, the convolution kernel is also normal density by the reproductivity of normal distribution. Since normal density functions are strictly totally positive of order ∞, the convolution kernel $K_h^*(u = x_j - x_i)$ is always positive definite [Karlin (1968)].

Remark 2 : The condition of symmetry might be relaxed as long as the convolution kernel $K_h^*(u = x_j - x_i)$ is symmetric. The example of the Gumbel kernel is demonstrated in section 2.4.

2.2.3 The discretized LSMDE (DLSMDE)

The implementation of kernel-based nonparametric curve estimators is considered computationally slow in comparison to other methods. Even though computational

speeds are improving rapidly, there still exists need for fast implementation of kernel density estimates. Since the nonparametric density estimation is one of the building blocks for exploratory data analysis, especially dynamic graphics, speed improvement becomes an important part of the study. One of the several attempts to accelerate the computation of smoothing process are the so called binning methods suggested by a number of authors [Härdle and Scott (1992), Härdle (1990), and Fan and Marron (1994)]. The idea behind binning methods is to reduce the number of kernel evaluations which occupy most of the computing time, exploiting the fact that many of the data points are close to each other. The data are approximated by equally spaced data (bins or regular mesh points). A naive implementation of binning methods usually reduces the number of kernel evaluations from $O(n \cdot m)$ to $O(m^2)$ where m is the number of the equally spaced mesh points [Fan and Marron (1994)].

In the computational speed sense, we note that the optimized weight of the ordinary LSMDE is the solution of the quadratic programming of size n. If we introduce m regular mesh points for the data set of size n and if m is much less than n, the size of optimization problem reduces significantly. Since the quadratic programming is known to be computationally complex, the advantage of the binning method lies rather in reducing the size of the optimization problem than in reducing the number of kernel evaluations.

Binning methods have another possible advantage for the stability of the LSMDE in addition to the speed improvement. Even if C is proven to be positive definite, there still exists a possibility that some of data points cluster at some locations (for example, the modes of the true density). We note that the (i, j)-th element of C in (2.15) is $K_h^*(x_i - x_j)$. If x_i and x_j are close to each other, then $K_h^*(x_i - x_j)$ becomes close to

$K_h^*(0)$, diagonal elements of C. This indicates that C might be near-singular and make the quadratic programming problem computationally indefinite. The binning methods enable us to maintain the positive definiteness of C computationally by separating the elements of C in a stable manner.

A simple form of binning method is to introduce the regular mesh points by replacing X_i by the nearest mesh point $x_{j(i)}$ and to do the estimation using these modified regular mesh points. While the ordinary LSMDE assigns positive weights to the actual data points, the discretized estimator assigns positive weights to the regular mesh points.

The modification is as follows:

STEP 1 : Generate regular mesh points

Divide the range of data into m equally spaced mesh points, g_j for j=1,...,m such that $g_j = g_1 + (j-1)s$, for j=1,...,m where s is a fixed grid width of mesh points.

STEP 2 : Define the least squares kernel density estimate as

$$\hat{f}_D(y) = \sum_{j=1}^{m} b_j K_h(y - g_j) \tag{2.23}$$

STEP 3 : Solve the following modified QP problem for b_j, j=1,...,m

$$\min_{\underline{b}} \quad \hat{Q}(\hat{f}_D; \underline{b}) = \underline{b}^T \underline{d}_2 + \frac{1}{2} \underline{b}^T C_2 \underline{b}$$

$$\text{s.t.} \quad \underline{1}^T \underline{b} = 1$$

$$I_m \underline{b} \geq \underline{0} \tag{2.24}$$

where $\underline{b}$ is (m x 1) column vector of unknowns,

$\underline{d}_2$ is (m x 1) column vector of

$$-\frac{2}{n}\left[\sum_{i=1}^{n} K_h(x_i - g_1), \quad \sum_{i=1}^{n} K_h(x_i - g_2), \quad \cdots, \quad \sum_{i=1}^{n} K_h(x_i - g_m)\right]^T \tag{2.25}$$

C_2 is (m x m) symmetric matrix of

$$2\begin{bmatrix} K_h^*(0) & K_h^*(g_1 - g_2) & \cdots & K_h^*(g_1 - g_m) \\ K_h^*(g_2 - g_1) & K_h^*(0) & \cdots & K_h^*(g_2 - g_m) \\ \vdots & \vdots & \ddots & \vdots \\ K_h^*(g_m - g_1) & K_h^*(g_m - g_2) & \cdots & K_h^*(0) \end{bmatrix} \tag{2.26}$$

Note that

$$\begin{aligned} \int \hat{f}^2(y)dy &= \int \sum_{i=1}^{m}\sum_{j=1}^{m} b_i b_j K_h(y - g_i)K_h(y - g_j)\, dy \\ &= \sum_{i=1}^{m}\sum_{j=1}^{m} b_i b_j \int K_h(y - g_i)K_h(y - g_j)dy \\ &= \sum_{i=1}^{m}\sum_{j=1}^{m} b_i b_j K_h^*(g_i - g_j) \end{aligned}$$

only depends on regular mesh points and the smoothing parameter h, not data points.

2.2.4 The LSMDE with normal kernels

Numerical implementations of the LSMDE and the discretized versions are performed using IMSL library routine QPROG (IMSL/MATH), a public domain

FORTRAN subroutine QLD.F, and the S-plus function *lsm* is implemented for graphic display of the estimates (see Appendix A for the details). Although any kernel function can be chosen for the LSMDE, the standard normal density seems natural despite its computational inefficiency. We have a complete setup for the quadratic programming with

$$K_h(x_i - x_j) = \frac{1}{h}K\left(\frac{x_i - x_j}{h}\right) = \frac{1}{h\sqrt{2\pi}}\exp\left[-\frac{(x_i - x_j)^2}{2h^2}\right], \tag{2.27}$$

$$K_h^*(x_i - x_j) = \int K_h(y - x_i)K_h(y - x_j) = \frac{1}{2h\sqrt{\pi}}\exp\left[-\frac{(x_i - x_j)^2}{4h^2}\right]. \tag{2.28}$$

Note $K_h^*(\Delta_{ij} = x_i - x_j)$ is the density of $N(0,\ \sigma = \sqrt{2}h)$.

2.3 Behavior of the LSMDE

We apply and analyze the LSMDE for two common statistical situations ($N(0,1)$ and the mixture of 2 normal densities) and one real data set (Buffalo snow fall data). The LSMDE turns out to be successful in revealing the multimodality and decomposing the true density function with quite small number of mixtures of given kernel functions and estimated mixing weights.

2.3.1 $N(0,1)$

Let us analyze how the least squares decomposition estimator works for a plain density function. A random sample of size n = 100 is generated from the standard

normal density, $N(0,1)$ using S-plus [Becker, Chambers and Wilks (1988)]. Our main interest is in how fast and accurately the LSMDE locates the mode (or equivalently mean in this case) of the density function with optimized weight close to 1.

To investigate the performance of the estimate, we have tried a broad range of the smoothing parameter (h ranges from 0.2 to 1.6 by the increment of 0.2). Table 2-1-1 and Fig 2-1-1 (A, B) show the optimized weight and the resulting density estimates. As expected, the estimate is very spiky for small h (0.2, 0.4) with 14 positive weights. The estimates rapidly become smooth as h reaches 0.6 and 0.8; and relatively large weights concentrate on the mean of $N(0,1)$.

For h = 1.0, the estimate is almost identical with the standard normal density. The density function of $N(0,1)$ is decomposed into 2 almost standard normal densities $N(X_{51} = -0.073213, h^2 = 1)$ and $N(X_{52} = -0.004672, h^2 = 1)$ with mixing weights $a_{51} = 0.591938$ and $a_{52} = 0.408062$.

For h = 1.2, one of the 2 positive remaining weights (a_{51} = 0.808246 at X_{51} = -0.073213) dominates the mixing proportion (a_{51} getting close to 1) and finally for h > 1.2, the estimate becomes the density of normal distribution with mean (X_{51} = -0.073213) fixed and the standard deviation = h (1.4, 1.6). Therefore, after roughly locating the mode (or mean) of $N(0,1)$, the least square mixture decomposition estimate is totally determined by h, and the estimate underestimates the mode of the true density of $N(0,1)$ afterwards.

The discretized version of the LSMDE with the mesh points of size 100 (Table 2-1-2 with Fig 2-1-2), 50 (Table 2-1-3 with Fig 2-1-3), and 25 (Table 2-1-4 with Fig 2-1-4) seems very similar to the ordinary LSMDE. Note that the case with 100 regular mesh points does not improve the quality of the estimate much, and the resulting estimate

seems to be almost the same as the case of 50 and 25. This suggests that the discretized version of the LSMDE could considerably decrease the computational effort by reducing the size of the optimization problem, as well as alleviating the ill-conditioning problem in the matrix C. Table 2-1-5 shows the changes in the number of resulting positive weights. We comment on this in the next chapter.

Table 2-1-1 : Positive weights for the sample data set by LSMDE
Data : Random sample of size n=100 from $N(0,1)$
Optimized only with data points

ID	X	H=0.2	H=0.4	H=0.6	H=0.8	H=1.0	H=1.2	H=1.4	H=1.6
1	-2.14254	0.008571	0	0	0	0	0	0	0
2	-2.05817	0.013333	0	0	0	0	0	0	0
5	-1.67983	0	0.020459	0	0	0	0	0	0
6	-1.63142	0	0.093557	0.101145	0	0	0	0	0
7	-1.58748	0.084545	0	0	0	0	0	0	0
8	-1.47917	0	0	0	0.061044	0	0	0	0
15	-1.03289	0.005719	0	0	0	0	0	0	0
16	-0.96114	0.090959	0	0	0	0	0	0	0
32	-0.54619	0.064754	0.338441	0	0	0	0	0	0
33	-0.48209	0.162021	0	0	0	0	0	0	0
40	-0.34539	0	0	0.026588	0	0	0	0	0
41	-0.30204	0	0	0.539544	0.174494	0	0	0	0
42	-0.24789	0	0	0	0.363047	0	0	0	0
51	-0.07321	0	0	0	0	0.591938	0.808249	1	1
52	-0.00467	0.177048	0	0	0	0.408062	0.191751	0	0
60	0.178057	0	0.284545	0	0	0	0	0	0
61	0.217351	0	0.033964	0	0	0	0	0	0
69	0.385819	0.175765	0	0	0.016946	0	0	0	0
70	0.393391	0	0	0	0.384468	0	0	0	0
80	0.683428	0	0	0.111546	0	0	0	0	0
81	0.798671	0	0	0.221176	0	0	0	0	0
82	0.829963	0.045468	0	0	0	0	0	0	0
83	0.849847	0.009257	0	0	0	0	0	0	0
86	0.90965	0	0.01423	0	0	0	0	0	0
87	1.024926	0	0.207996	0	0	0	0	0	0
92	1.114495	0.142105	0	0	0	0	0	0	0
99	1.814402	0.008573	0	0	0	0	0	0	0
100	2.746376	0.011881	0.006809	0	0	0	0	0	0
#(positive weight)		14	8	5	5	2	2	1	1
Minimized Q[f,a]		-0.32588	-0.36145	-0.31213	-0.31040	-0.30831	-0.29511	-0.27674	-0.25750

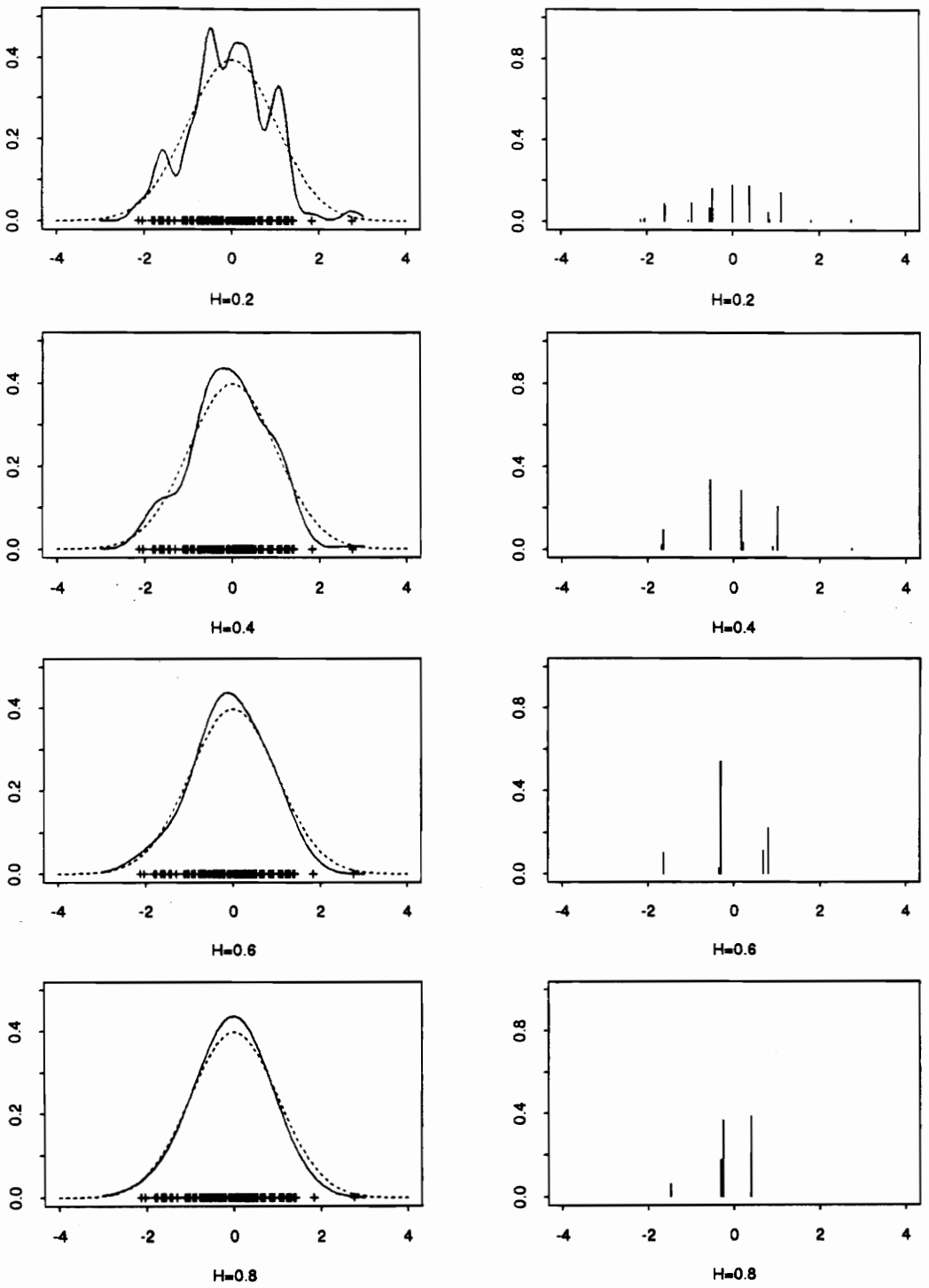


Figure 2-1-1 (A) : LSMDE for $N(0,1)$ with positive weights (n=100). The solid line is LSMDE and the dotted line is $N(0,1)$.

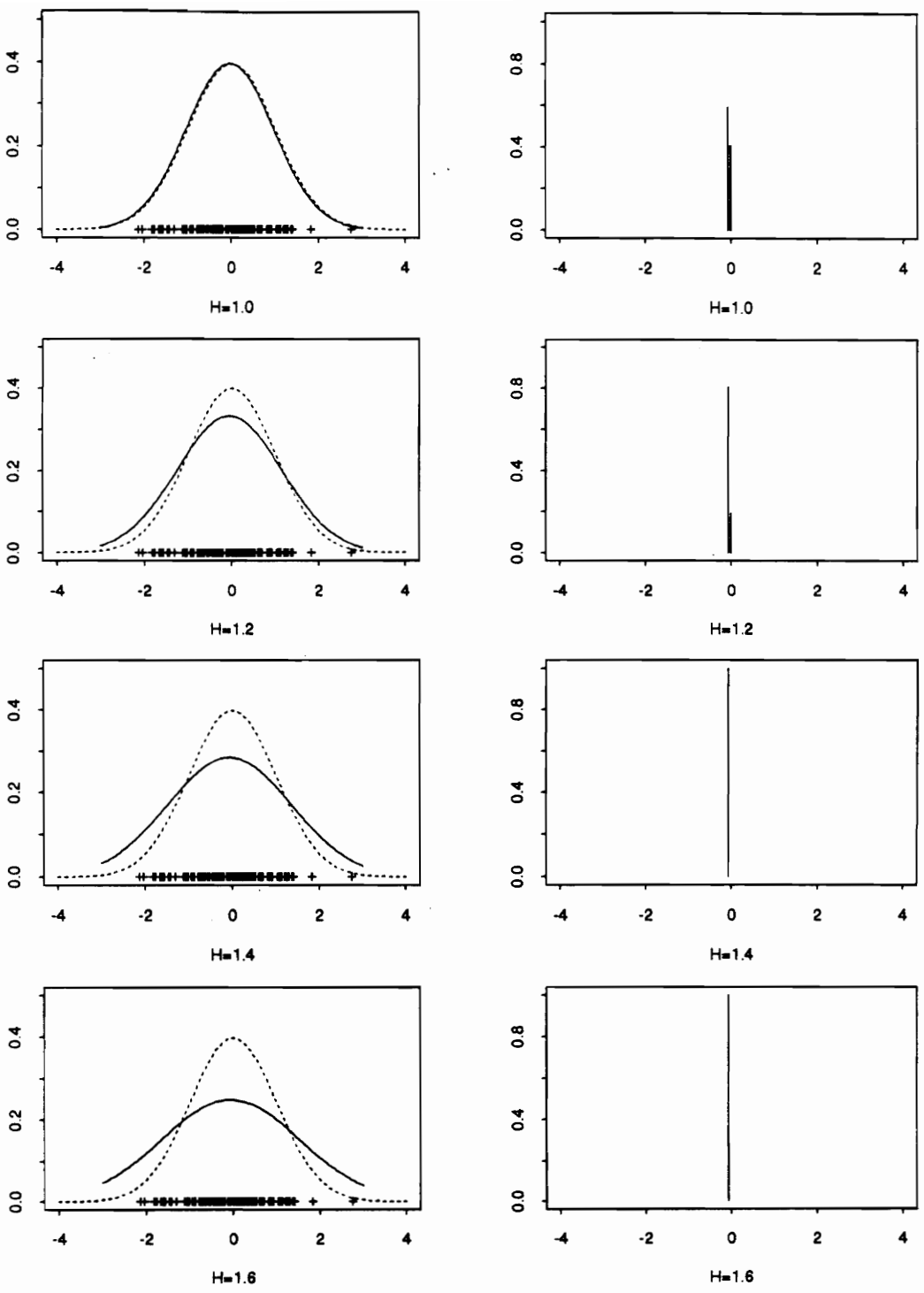


Figure 2-1-1 (B) : LSMDE for $N(0,1)$ with positive weights (n=100). The solid line is LSMDE and the dotted line is $N(0,1)$

Table 2-1-2 : Positive weights for the sample data set by DLSMDE

Data Random sample of size n=100 from N(0,1)

Optimized with 100 regular mesh points

ID	X	H=0.2	H=0.4	H=0.6	H=0.8	H=1.0	H=1.2	H=1.4	H=1.6
4	-2.1353	0.005813	0	0	0	0	0	0	0
5	-2.08289	0.014976	0	0	0	0	0	0	0
13	-1.66358	0	0.089062	0.007984	0	0	0	0	0
14	-1.61117	0.059457	0.023853	0.094332	0	0	0	0	0
15	-1.55875	0.024779	0	0	0	0	0	0	0
16	-1.50634	0	0	0	0.057445	0	0	0	0
17	-1.45393	0	0	0	0.000403	0	0	0	0
26	-0.98221	0.094534	0	0	0	0	0	0	0
34	-0.5629	0	0.279605	0	0	0	0	0	0
35	-0.51049	0.224489	0.055472	0	0	0	0	0	0
38	-0.35325	0	0	0.030101	0	0	0	0	0
39	-0.30083	0	0	0.534406	0.346836	0	0	0	0
40	-0.24842	0	0	0	0.176101	0	0	0	0
43	-0.09118	0	0	0	0	0.054124	0.192561	0.342445	0.463272
44	-0.03877	0.110155	0	0	0	0.945876	0.807439	0.657555	0.536728
45	0.013645	0.068449	0	0	0	0	0	0	0
48	0.170885	0	0.319884	0	0	0	0	0	0
51	0.328125	0.000809	0	0	0	0	0	0	0
52	0.380538	0.178493	0	0	0.419215	0	0	0	0
59	0.747431	0	0	0.278258	0	0	0	0	0
60	0.799845	0.02081	0	0.054919	0	0	0	0	0
61	0.852258	0.034954	0	0	0	0	0	0	0
63	0.957084	0	0.005165	0	0	0	0	0	0
64	1.009498	0	0.219979	0	0	0	0	0	0
66	1.114324	0.141719	0	0	0	0	0	0	0
80	1.84811	0.008833	0	0	0	0	0	0	0
97	2.739136	0.011731	0	0	0	0	0	0	0
98	2.791549	0	0.00698	0	0	0	0	0	0
#(positive weights)		15	8	6	5	2	2	2	2
Minimized Q[f,a]		-0.32599	-0.31616	-0.31215	-0.3104	-0.30833	-0.29513	-0.27673	-0.25748

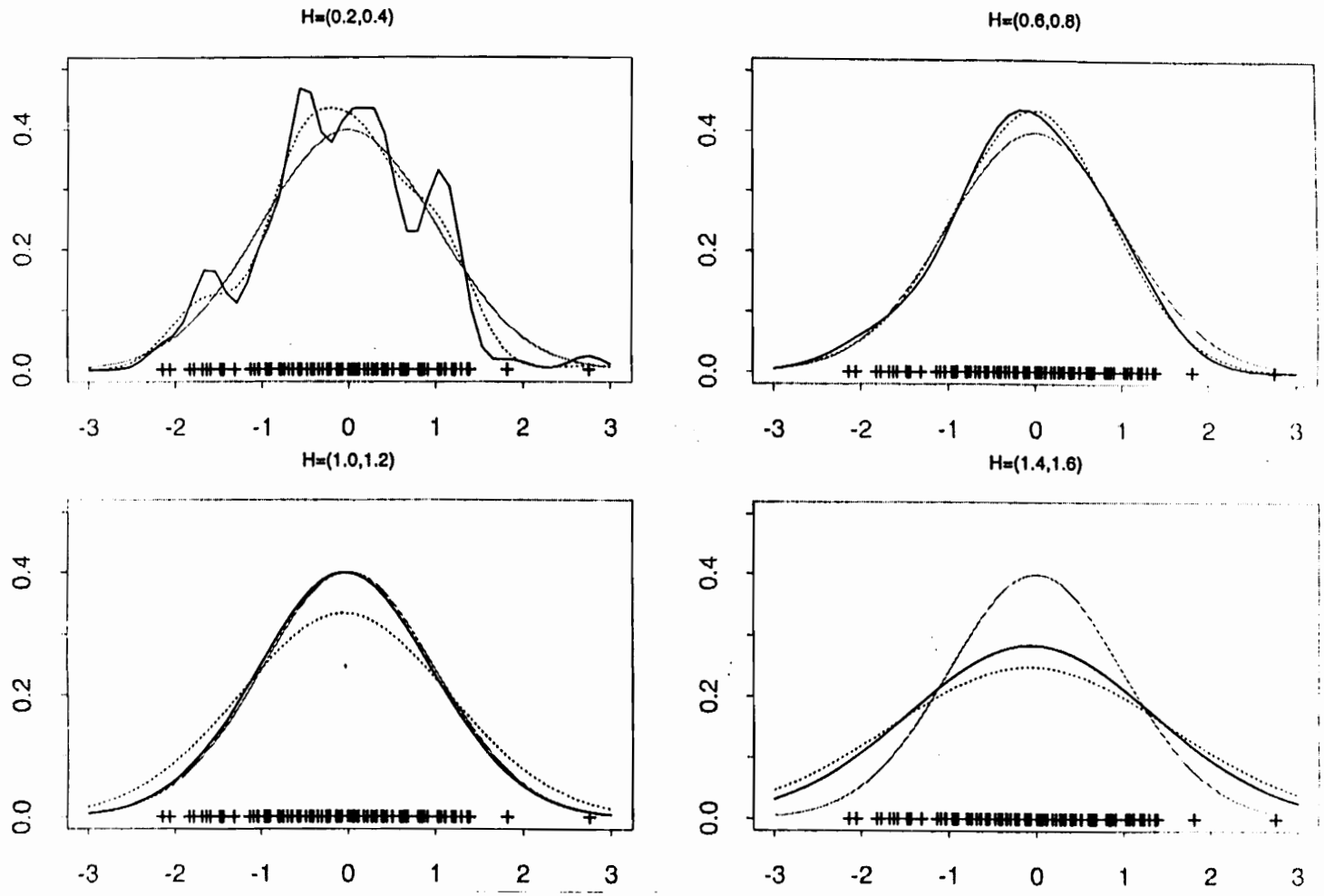


Figure 2-1-2 : DLSMDE for $N(0,1)$ (n=100 with 100 mesh points). The solid line corresponds to the smaller bandwidth. The dotted line corresponds to the larger bandwidth. The vague line is $N(0,1)$.

Table 2-1-3 : Positive weights for the sample data set by DLSMDE
Data : Random sample of size n=100 from $N(0,1)$
Optimized with 50 regular mesh points

ID	X	H=0.2	H=0.4	H=0.6	H=0.8	H=1.0	H=1.2	H=1.4	H=1.6
3	-2.08075	0.021071	0	0	0	0	0	0	0
7	-1.65716	0.029993	0.112048	0.067466	0	0	0	0	0
8	-1.55127	0.054716	0	0.035178	0.010829	0	0	0	0
9	-1.44537	0	0	0	0.050678	0	0	0	0
13	-1.02178	0.049401	0	0	0	0	0	0	0
14	-0.91589	0.048171	0	0	0	0	0	0	0
17	-0.5982	0.021211	0.213489	0	0	0	0	0	0
18	-0.4923	0.20183	0.117109	0	0	0	0	0	0
19	-0.38641	0	0	0.075783	0	0	0	0	0
20	-0.28051	0	0	0.496103	0.512953	0	0	0	0
22	-0.06872	0.084627	0	0	0	0.809439	1	1	1
23	0.037178	0.091403	0	0	0	0.190561	0	0	0
24	0.143074	0	0.291693	0	0	0	0	0	0
25	0.24897	0	0.027219	0	0	0	0	0	0
26	0.354867	0.14448	0	0	0.354255	0	0	0	0
27	0.460763	0.032284	0	0	0.071285	0	0	0	0
29	0.672555	0	0	0.012932	0	0	0	0	0
30	0.778452	0.046425	0	0.312539	0	0	0	0	0
32	0.990244	0	0.231021	0	0	0	0	0	0
33	1.09614	0.146054	0	0	0	0	0	0	0
34	1.202037	0.007427	0	0	0	0	0	0	0
40	1.837414	0.0091	0	0	0	0	0	0	0
48	2.684584	0.003022	0	0	0	0	0	0	0
49	2.79048	0.008782	0.00742	0	0	0	0	0	0
#(positive weight)		17	7	6	5	2	1	1	1
Minimized Q[f,a]		-0.32559	-0.31611	-0.31213	-0.31039	-0.3083	-0.29512	-0.27674	-0.2575

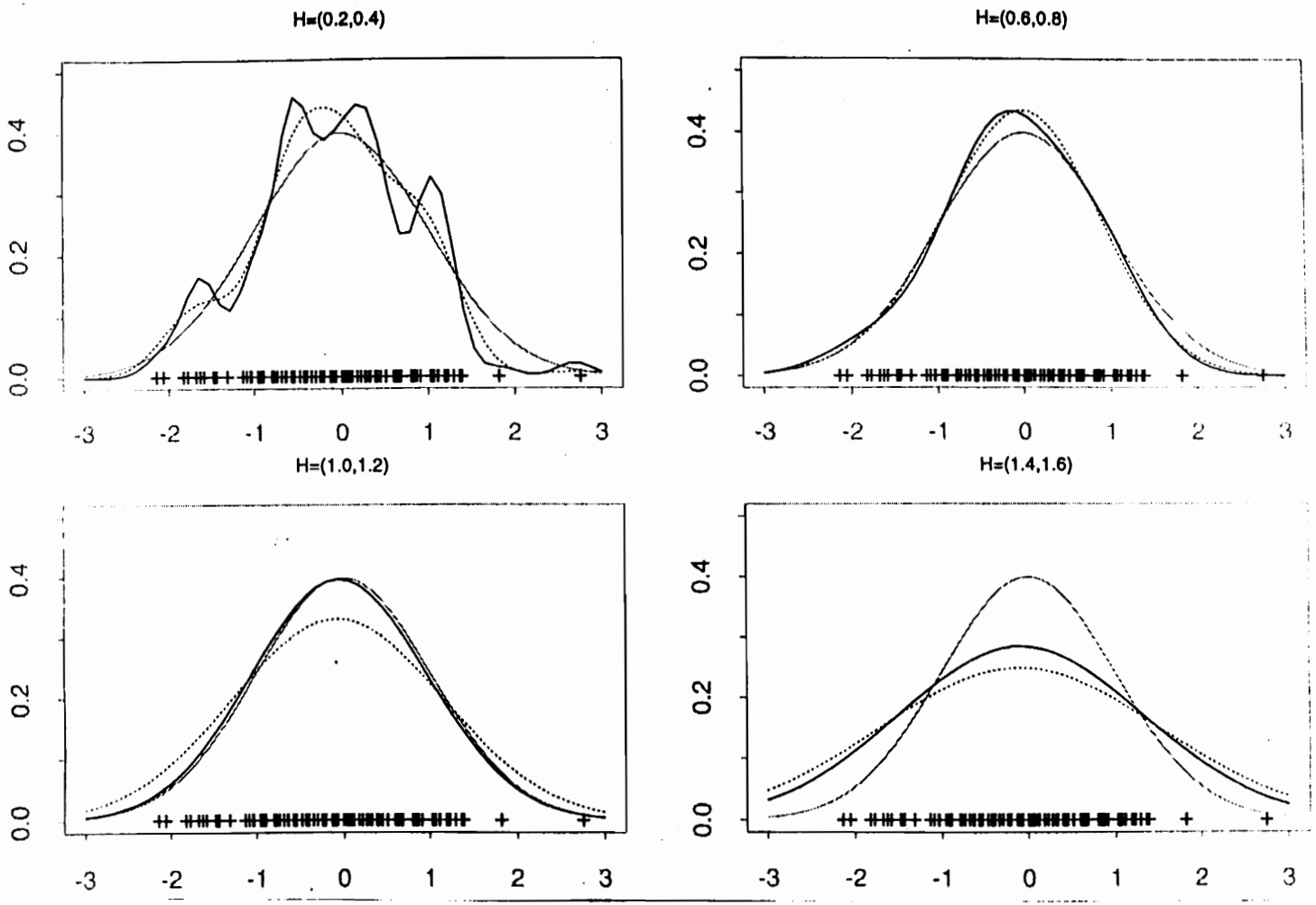


Figure 2-1-3 : DLSMDE for $N(0,1)$ (n=100 with 50 mesh points). The solid line corresponds to the smaller bandwidth. The dotted line corresponds to the larger bandwidth. The vague line is $N(0,1)$.

Table 2-1-4 : Positive weights for the sample data set by DLSMDE
Data : Random sample of size n=100 from N(0,1)
Optimized with 25 regular mesh points

ID	X	H=0.2	H=0.4	H=0.6	H=0.8	H=1.0	H=1.2	H=1.4	H=1.6
2	-2.07633	0.017568	0	0	0	0	0	0	0
4	-1.64392	0.072066	0.114487	0.096171	0.014377	0	0	0	0
5	-1.42772	0.01497	0	0	0.040877	0	0	0	0
7	-0.99531	0.082973	0	0	0	0	0	0	0
9	-0.5629	0.19238	0.318399	0	0	0	0	0	0
10	-0.3467	0.030946	0	0.504041	0.44588	0	0	0	0
11	-0.13049	0.10777	0	0.056123	0	0.617257	0.707945	0.79108	0.85794
12	0.085714	0.073814	0.256459	0	0	0.382743	0.292055	0.20892	0.14206
13	0.301919	0.127851	0.065308	0	0.4524	0	0	0	0
14	0.518123	0.073276	0	0	0.046466	0	0	0	0
15	0.734328	0	0	0.343665	0	0	0	0	0
16	0.950533	0.088978	0.220502	0	0	0	0	0	0
17	1.166738	0.098236	0.01681	0	0	0	0	0	0
20	1.815352	0.007647	0	0	0	0	0	0	0
24	2.680171	0.010989	0.005365	0	0	0	0	0	0
25	2.896376	0.000536	0.00267	0	0	0	0	0	0
#(positive weights)		15	8	4	5	2	2	2	2
Minimized Q[f,a]		-0.32471	-0.31591	-0.31211	-0.31039	-0.30811	-0.29478	-0.27643	-0.25725

Table 2-1-5 : Number of positive weights for the sample data set by LSMDE
Data : Random sample of size n=100 from N(0,1)

h	LSMDE	DLSMDE(100)	DLSMDE(50)	DLSMDE(25)
0.2	**14**	**15**	**17**	**15**
0.4	**8**	**8**	**7**	**8**
0.6	**5**	**6**	**6**	**4**
0.8	**5**	**5**	**5**	**5**
1.0	**2**	**2**	**2**	**2**
1.2	**2**	**2**	**1**	**2**
1.4	**1**	**2**	**1**	**2**
1.6	**1**	**2**	**1**	**2**

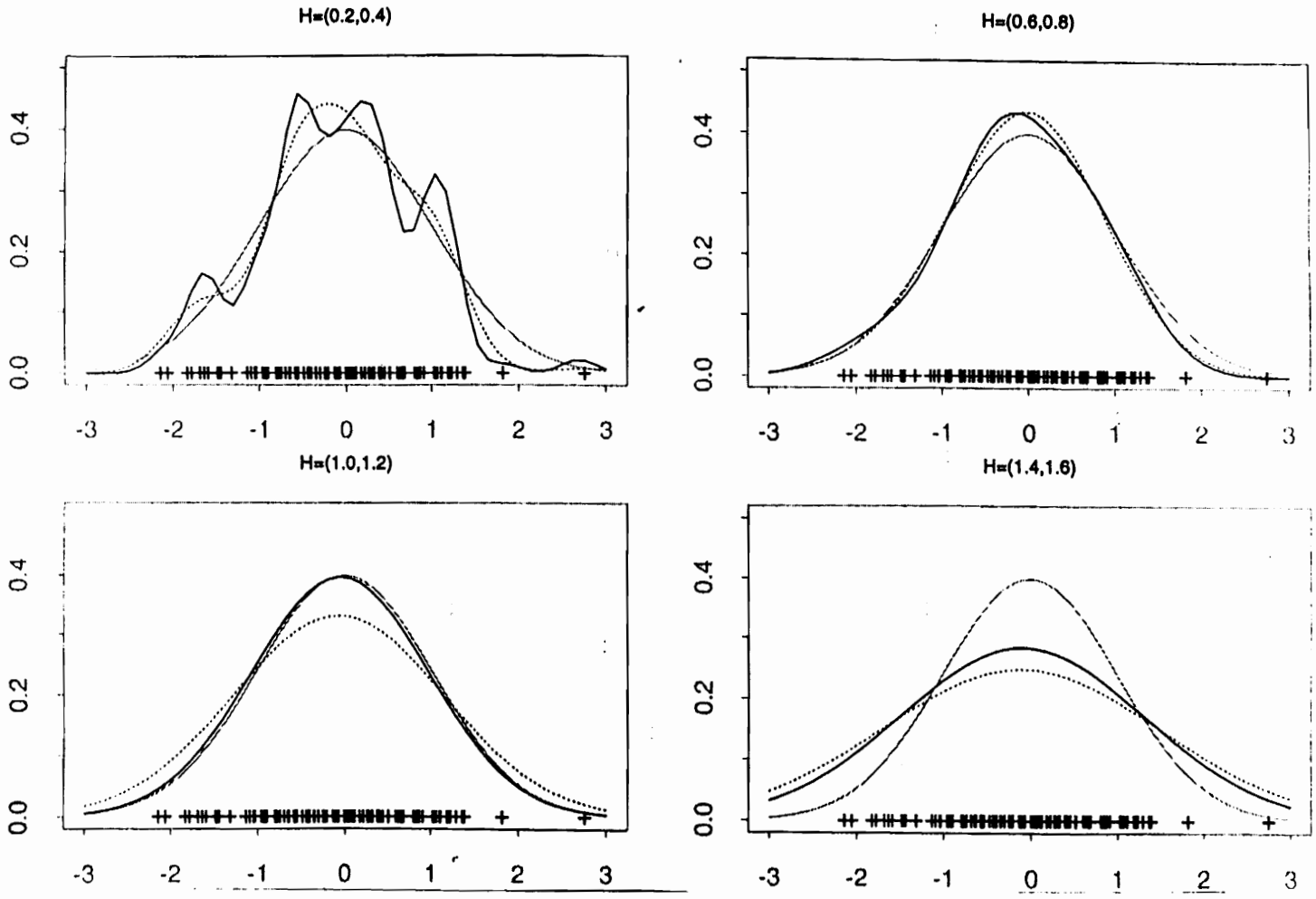


Figure 2-1-4 : DLSMDE for $N(0,1)$ (n=100 with 25 mesh points).The solid line corresponds to the smaller bandwidth. The dotted line corresponds to the larger bandwidth. The vague line is $N(0,1)$.

2.3.2 Mixture of two normal densities

We choose a mixture of 2 normal density functions which is bimodal in order to see how the LSMDE locates true modes and estimates mixing weights. We generate a random sample of size n=50 from the mixture of $N(-1, 1)$ and $N(2, 1)$ with weights 0.6 and 0.4 respectively using MINITAB. This mixture has the probability density function

$$f(y) = 0.6\frac{1}{\sqrt{2\pi}}\exp\left(-\frac{(y+1)^2}{2}\right) + 0.4\frac{1}{\sqrt{2\pi}}\exp\left(-\frac{(y-2)^2}{2}\right) \qquad (2.29)$$

and has two modes near component means.

For the same smoothing parameter choice of the previous example, resulting weights and estimates are summarized in Table 2-2-1 and Fig 2-2-1 (A, B).

Like the ordinary kernel density estimate, the LSMDE shows the trade-off relationship between variance and bias by getting smoother as h grows. For smaller value of h (0.2, 0.4), the estimate is very spiky, showing multiple modes. However, only 20 (10 when h = 0.4)) data points out of 50 have positive weights. Most data points have zero weight. For h = 0.6, 0.8, we observe that the LSMDE is much smoother as h increases, and locates the two modes of the true density pretty well. The number of positive weights gets smaller (7 for h = 0.6, 5 for h = 0.8).

The best fit is when h = 1.0. The estimate is almost identical to the true density at both edges of the true density and shows two modes prominently — over-estimated around X = -1.0, under-estimated around X = 2.0. Furthermore, the optimized weights concentrate at the data points close to the true modes of the underlying density and seem to reflect the mixing weights of the true density for this example — $a_{16} = 0.591794$ at

$X_{16} = -0.97925$and $a_{43} = 0.372782$at $X_{43} = 2.15867$. The number of positive weights is 3 with one weight relatively small ($a_{17} = 0.035424$) close to the one of the true mode (at $X_{17} = -0.87457$). This implies that the LSMDE decomposes the underlying density with a mixture of 3 normal densities (or 2 significant normal densities).

For h=1.2, the positions of the significant weights stay close to the true modes. The estimates, however, seem over-smoothed, and the mode at $X = 2$ is quite under-estimated. Finally as h increases to 1.6, the estimate is unimodal. Positive weights, nevertheless, still stay close to the location of true modes.

Let us examine the discretized LSMDE for this mixture density. Numbers of regular mesh points of 100 (Table 2-2-2 with Fig 2-2-2), 50 (Table 2-2-3 with Fig 2-2-3) and 25 (Table 2-2-4 with Fig 2-2-4) have been tried. The discretized estimates are almost identical to the LSMDE. The behavior of the discretized estimates is almost the same as the ordinary LSMDE. The number of the positive weights decreases rapidly as h increases and the locations of the significant weights are close to the true modes of the underlying density. Again we find that the case of 100 regular mesh points does not improve the quality of the estimate much and the resulting estimate seems to be almost the same as the case of 50 and 25.

We can make several comments about the examples we just examined. First, the number of positive weights is much smaller than the sample size unless h is small enough to cover only a few data points. The LSMDE could be used to decompose the true density with the finite mixture of given kernel functions.

Second, the objective function, $\hat{Q}[\hat{f}, \underline{a}]$ always has smaller value as $h \to 0$. We see that the determination of the optimal smoothing parameter through the global minimization of the objective function over h is meaningless.

Third, the number of remaining positive weights tends to be smaller as $h \to \infty$, but *not always*. For the $N(0,1)$ example, the number of positive weights for all the cases always decreases as h increases (Table 2-1-5). Table 2-2-5, however, shows that the number of positive weights increases from 3 (when h = 1.2 of the first column) to 4 (when h = 1.4 of the first column). Silverman (1981) showed that the number of modes is a decreasing function of the smoothing parameter for certain kernel function including the standard normal kernel; and used the ordinary kernel estimator to test the multimodality of the underlying density. This property has been used by Minotte and Scott (1993) to construct a new graphical method called the mode tree for investigating the mode. On the other hand, a similar phenomenon has not happened with the LSMDE. While these methods are focusing on the detection of modes, the LSMDE concentrates on the identification of the components or clusters embedded in the data set.

Also, the location of the large weights does not stay at certain fixed points as h changes. It rather seems that significant weights tend to converge to some few data points, eventually to one data point with weight 1. Tables of positive weights give the rough idea of how possible clusters change in locations graphically.

From a computational point of view, the discretized version of the LSMDE gives us almost the same quality of estimator and alleviates the ill-condition problem in the quadratic programming and furthermore relieves computational effort for a moderate choice of the number of regular mesh points.

Table 2-2-1 : Positive weights for the sample data set by LSMDE

Data : Random sample of size n=50 from 0.6 N(-1,1) + 0.4 N(2,1)

Optimized without mesh points

ID	X	H=0.2	H=0.4	H=0.6	H=0.8	H=1.0	H=1.2	H=1.4	H=1.6
1	-3.41546	0.018517	0.033104	0.027179	0.025165	0	0	0	0
2	-3.01528	0.018465	0.005748	0	0	0	0	0	0
5	-1.90267	0.058973	0	0	0	0	0	0	0
6	-1.80615	0.07057	0	0	0	0	0	0	0
8	-1.64736	0	0.178165	0	0	0	0	0	0
9	-1.48513	0	0	0.260495	0	0	0	0	0
11	-1.37668	0.015869	0	0	0	0	0	0	0
12	-1.24968	0.119102	0.103981	0	0	0	0	0	0
16	-0.97925	0	0	0	0.317056	0.591794	0.644084	0.090426	0
17	-0.87457	0	0	0	0.274688	0.035424	0	0.583985	0
18	-0.81565	0.069792	0	0	0	0	0	0	0.466264
19	-0.71933	0.045365	0	0	0	0	0	0	0.244561
21	-0.62934	0	0	0.057887	0	0	0	0	0
22	-0.48066	0	0	0.269622	0	0	0	0	0
23	-0.42169	0	0.284048	0	0	0	0	0	0
26	-0.26769	0.134102	0	0	0	0	0	0	0
27	-0.22444	0.033915	0	0	0	0	0	0	0
30	0.27425	0.004377	0	0	0	0	0	0	0
31	0.64311	0.012882	0	0	0	0	0	0	0
32	0.89073	0.037484	0	0	0	0	0	0	0
33	1.07768	0	0.072763	0	0	0	0	0	0
35	1.44265	0.059858	0	0	0	0	0	0	0
36	1.50842	0.030956	0	0	0	0	0	0	0
37	1.60814	0	0	0.138149	0	0	0	0	0
38	1.72004	0	0.088821	0.08048	0	0	0	0	0
39	1.96977	0	0.086082	0	0.282535	0	0	0	0
40	2.05421	0.081448	0	0	0	0	0	0	0
41	2.10262	0.042845	0	0	0	0	0	0	0
43	2.15867	0	0	0	0	0.372782	0.350195	0.290878	0.289175
44	2.65131	0	0	0	0.100556	0	0.005721	0.034712	0
46	2.80147	0.053216	0.004666	0.166189	0	0	0	0	0
47	2.93491	0.077468	0.142624	0	0	0	0	0	0
50	3.55403	0.014796	0	0	0	0	0	0	0
#(positive weights)		20	10	7	5	3	3	4	3
Minimized Q[f,a]		-0.21987	-0.19282	-0.18515	-0.18234	-0.17728	-0.16809	-0.16022	-0.15477

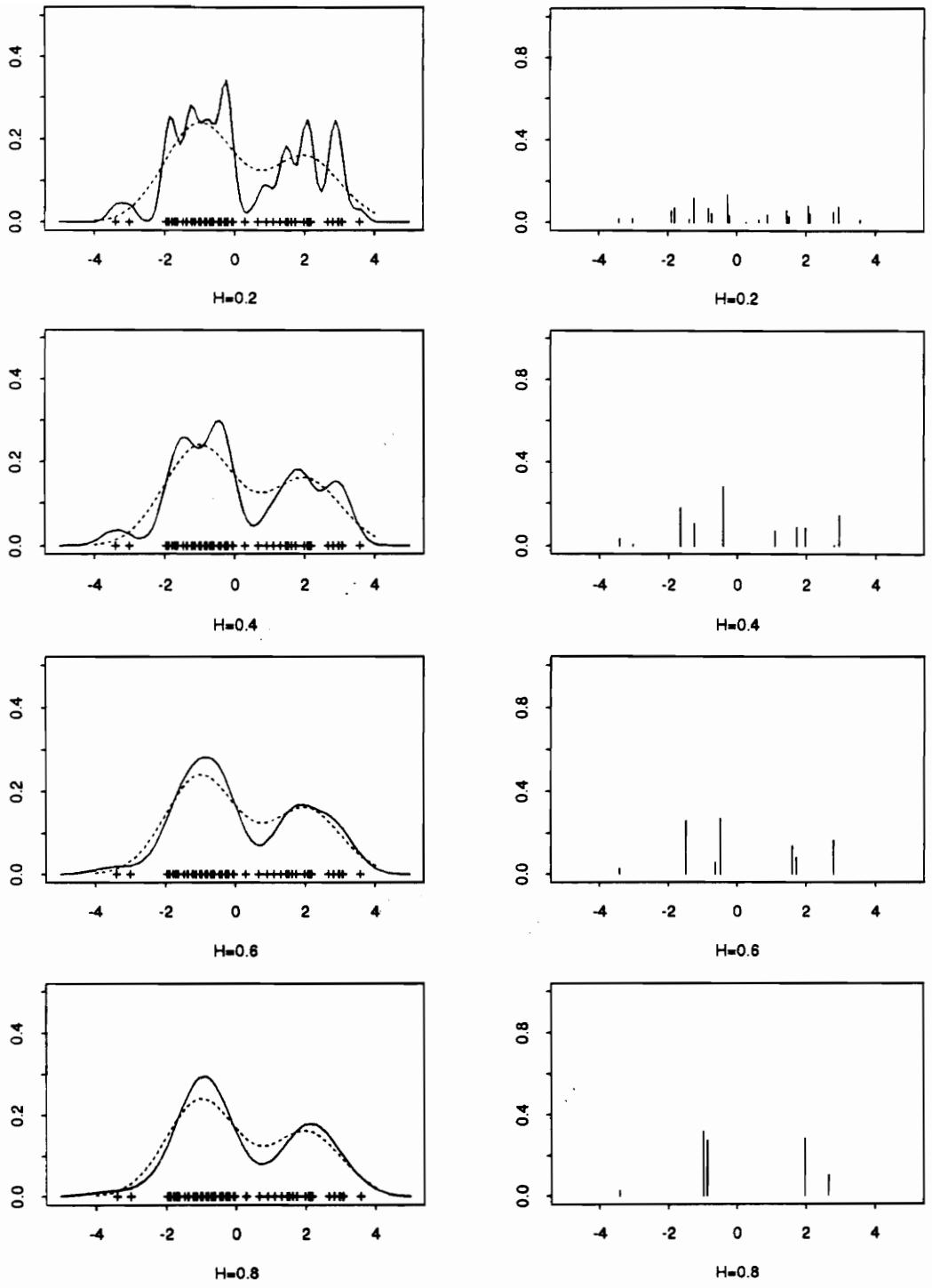


Figure 2-2-1 (A) : LSMDE for 0.6 $N(-1,1)$ + 0.4 $N(2,1)$ with positive weights (n=50). The solid line is LSMDE and the dotted line is 0.6 $N(-1,1)$ + 0.4 $N(2,1)$.

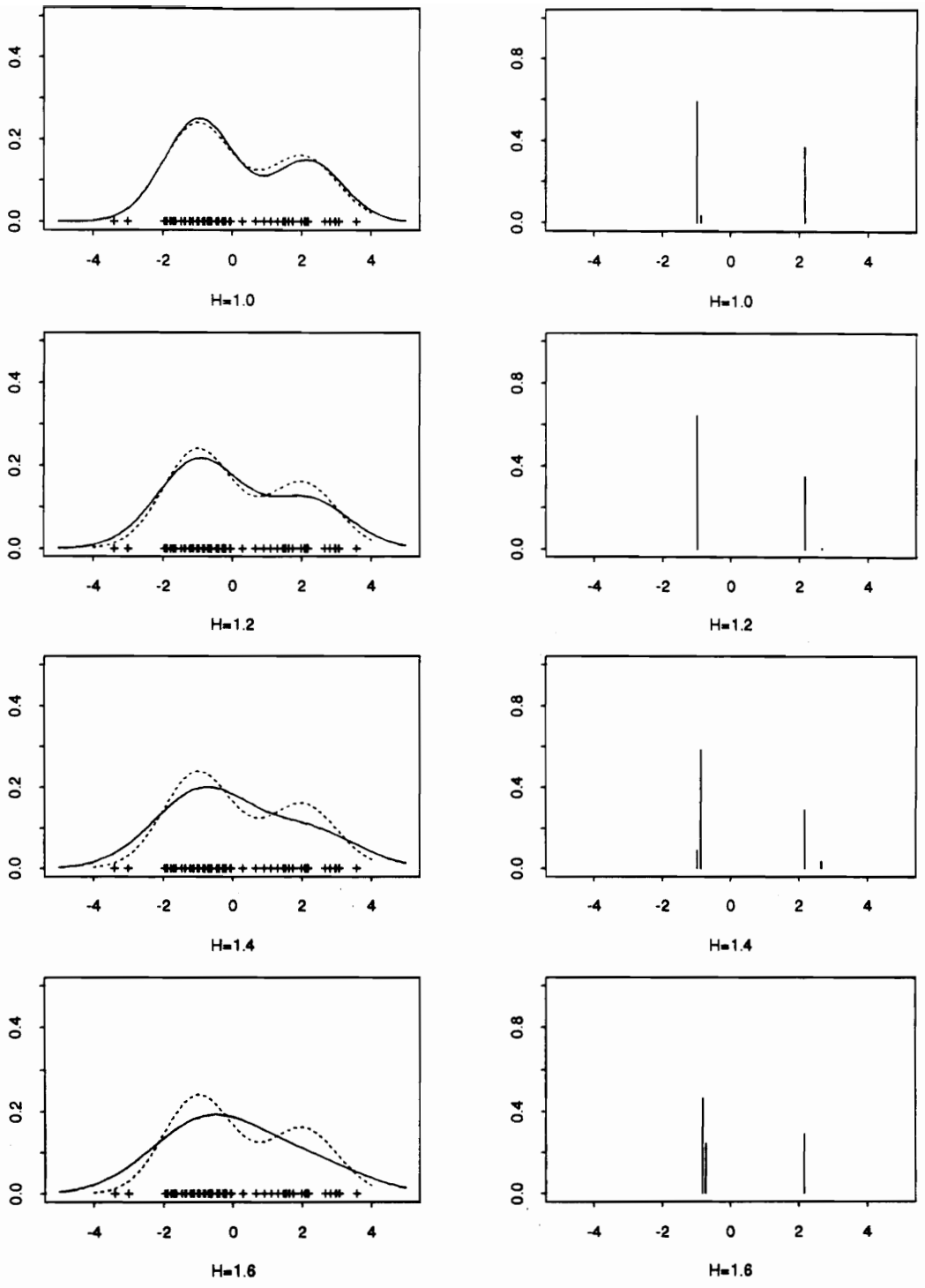


Figure 2-2-1 (B) : LSMDE for 0.6 $N(-1,1)$ + 0.4 $N(2,1)$ with positive weights (n=100). The solid line is LSMDE and the dotted line is 0.6 $N(-1,1)$ + 0.4 $N(2,1)$.

Table 2-2-2 : Positive weights for the sample data set by DLSMDE

Data : Random sample of size n=50 from 0.6 N(-1,1) + 0.4 N(2,1)

Optimized with 100 regular mesh points

ID	Mesh pt	H=0.2	H=0.4	H=0.6	H=0.8	H=1.0	H=1.2	H=1.4	H=1.6
2	-3.49203	0	0	0	0.025195	0	0	0	0
3	-3.4186	0.016641	0	0.02693	0	0	0	0	0
5	-3.27174	0	0.040376	0	0	0	0	0	0
8	-3.05146	0.018784	0	0	0	0	0	0	0
24	-1.87659	0.108154	0	0	0	0	0	0	0
25	-1.80316	0.017592	0	0	0	0	0	0	0
27	-1.6563	0	0.076633	0	0	0	0	0	0
28	-1.58287	0	0.131388	0	0	0	0	0	0
29	-1.50944	0	0	0.198914	0	0	0	0	0
30	-1.43601	0	0	0.057201	0	0	0	0	0
32	-1.28916	0.135293	0	0	0	0	0	0	0
34	-1.1423	0	0.080346	0	0	0	0	0	0
36	-0.99544	0	0	0	0.079386	0.232666	0.06737	0	0
37	-0.92201	0	0	0	0.511777	0.397454	0.585264	0.252459	0
38	-0.84858	0.013104	0	0	0	0	0	0.428442	0
39	-0.77515	0.105402	0	0	0	0	0	0	0.466904
40	-0.70172	0	0	0	0	0	0	0	0.257284
42	-0.55486	0	0	0.170606	0	0	0	0	0
43	-0.48143	0	0	0.160123	0	0	0	0	0
44	-0.40801	0	0.277697	0	0	0	0	0	0
46	-0.26115	0.168293	0	0	0	0	0	0	0
55	0.399716	0.002523	0	0	0	0	0	0	0
56	0.473145	0.012774	0	0	0	0	0	0	0
61	0.840291	0.004766	0	0	0	0	0	0	0
62	0.913721	0.038335	0	0	0	0	0	0	0
64	1.060579	0	0.07313	0	0	0	0	0	0
69	1.427725	0.019844	0	0	0	0	0	0	0
70	1.501154	0.071313	0	0	0	0	0	0	0
72	1.648012	0	0	0.220473	0	0	0	0	0
74	1.794871	0	0.094998	0	0	0	0	0	0
75	1.8683	0	0.074005	0	0.001995	0	0	0	0
76	1.941729	0	0	0	0.263383	0	0	0	0
78	2.088588	0.122824	0	0	0	0	0	0	0
79	2.162017	0	0	0	0	0.012001	0	0	0
80	2.235446	0	0	0	0	0.357879	0.050487	0	0
81	2.308875	0	0	0	0	0	0.296879	0.319099	0.275812
85	2.602592	0	0	0	0.118264	0	0	0	0
87	2.74945	0	0	0.011891	0	0	0	0	0
88	2.822879	0.013897	0	0.153862	0	0	0	0	0
89	2.896309	0.114974	0.151428	0	0	0	0	0	0
99	3.630601	0.015487	0	0	0	0	0	0	0
#(positive weights)		18	9	8	6	4	4	3	3
Minimized Q[f,a]		-0.22099	-0.19314	-0.18517	-0.18235	-0.17737	-0.16829	-0.16037	-0.15483

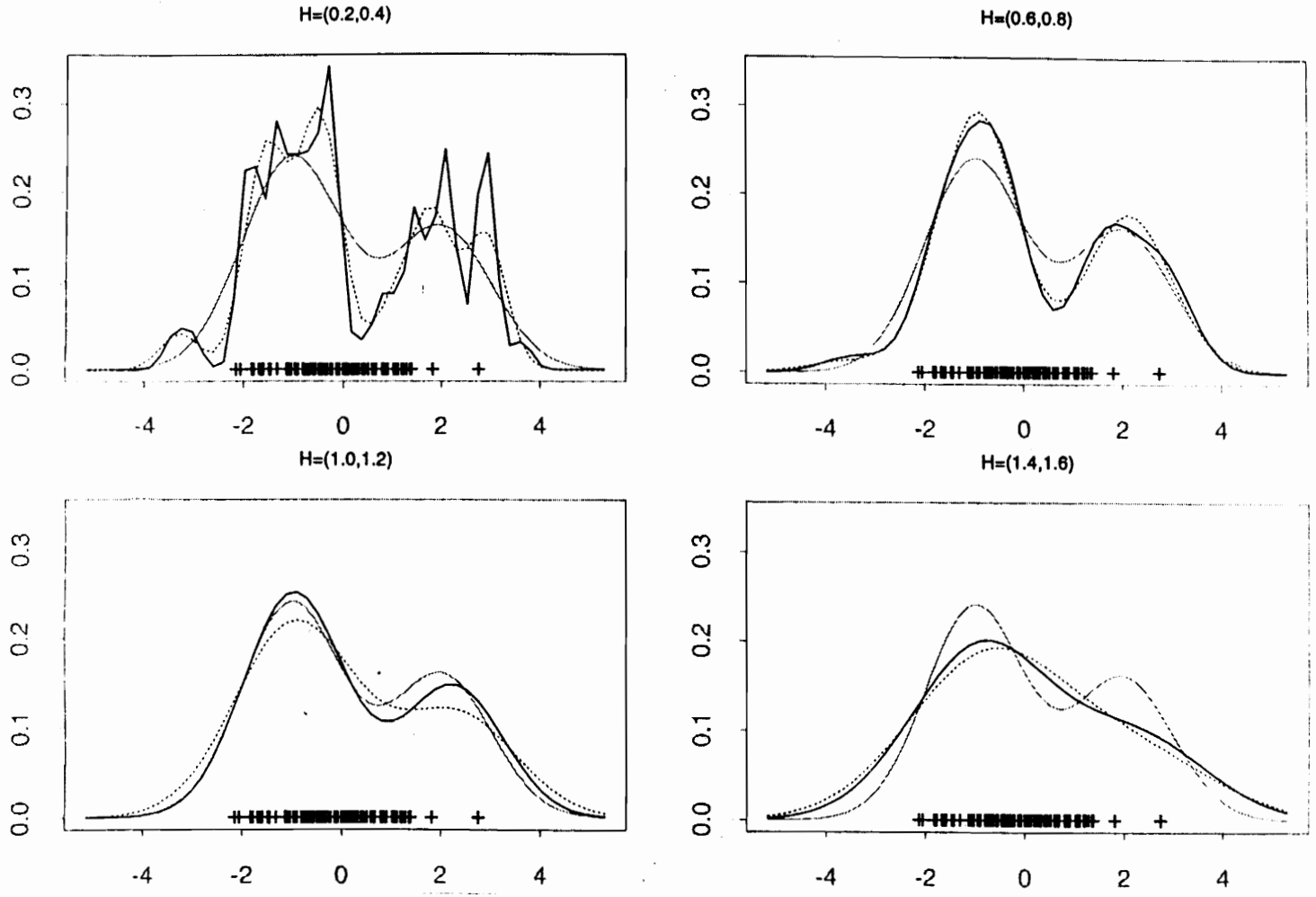


Figure 2-2-2 : DLSMDE for 0.6 $N(-1,1)$ + 0.4 $N(2,1)$ (n=100 with 100 mesh points). The solid line corresponds to the smaller bandwidth. The dotted line corresponds to the larger bandwidth.

Table 2-2-3 : Positive weights for the sample data set by DLSMDE

Data : Random sample of size n=50 from 0.6 N(-1,1) + 0.4 N(2,1)

Optimized with 50 regular mesh points

ID	Mesh pt	H=0.2	H=0.4	H=0.6	H=0.8	H=1.0	H=1.2	H=1.4	H=1.6
1	-3.56546	0	0	0	0.009454	0	0	0	0
2	-3.4171	0.018231	0	0.027198	0.015245	0	0	0	0
3	-3.26875	0	0.040349	0	0	0	0	0	0
4	-3.12039	0.005043	0	0	0	0	0	0	0
5	-2.97203	0.013225	0	0	0	0	0	0	0
12	-1.93353	0.057767	0	0	0	0	0	0	0
13	-1.78518	0.071016	0	0	0	0	0	0	0
14	-1.63682	0	0.1669	0	0	0	0	0	0
15	-1.48846	0	0.041396	0.255888	0	0	0	0	0
16	-1.34011	0.068296	0	0	0	0	0	0	0
17	-1.19175	0.070801	0.071066	0	0	0	0	0	0
18	-1.04339	0	0	0	0.146728	0.228536	0.151548	0	0
19	-0.89504	0.016922	0	0	0.444916	0.401626	0.500639	0.67792	0
20	-0.74668	0.092463	0	0	0	0	0	0	0.723927
21	-0.59832	0	0	0.153595	0	0	0	0	0
22	-0.44996	0	0.240895	0.177152	0	0	0	0	0
23	-0.30161	0.127013	0.044723	0	0	0	0	0	0
24	-0.15325	0.043505	0	0	0	0	0	0	0
28	0.440177	0.012701	0	0	0	0	0	0	0
31	0.885248	0.044182	0	0	0	0	0	0	0
32	1.033605	0	0.069598	0	0	0	0	0	0
35	1.478676	0.090141	0	0	0	0	0	0	0
36	1.627033	0	0	0.192907	0	0	0	0	0
37	1.77539	0	0.12722	0.027113	0	0	0	0	0
38	1.923747	0	0.045363	0	0.25261	0	0	0	0
39	2.072104	0.122972	0	0	0	0	0	0	0
40	2.220461	0	0	0	0	0.369838	0.199588	0.163971	0.12652
41	2.368818	0	0	0	0	0	0.148225	0.158108	0.149553
42	2.517175	0	0	0	0.084314	0	0	0	0
43	2.665532	0	0	0	0.046733	0	0	0	0
44	2.813889	0.081093	0.062965	0.166148	0	0	0	0	0
45	2.962246	0.049635	0.089526	0	0	0	0	0	0
49	3.555673	0.014993	0	0	0	0	0	0	0
#(positive weights)		18	11	7	7	3	4	3	3
Minimized Q[f,a]		-0.21879	-0.19294	-0.18514	-0.18234	-0.17732	-0.16822	-0.16035	-0.15483

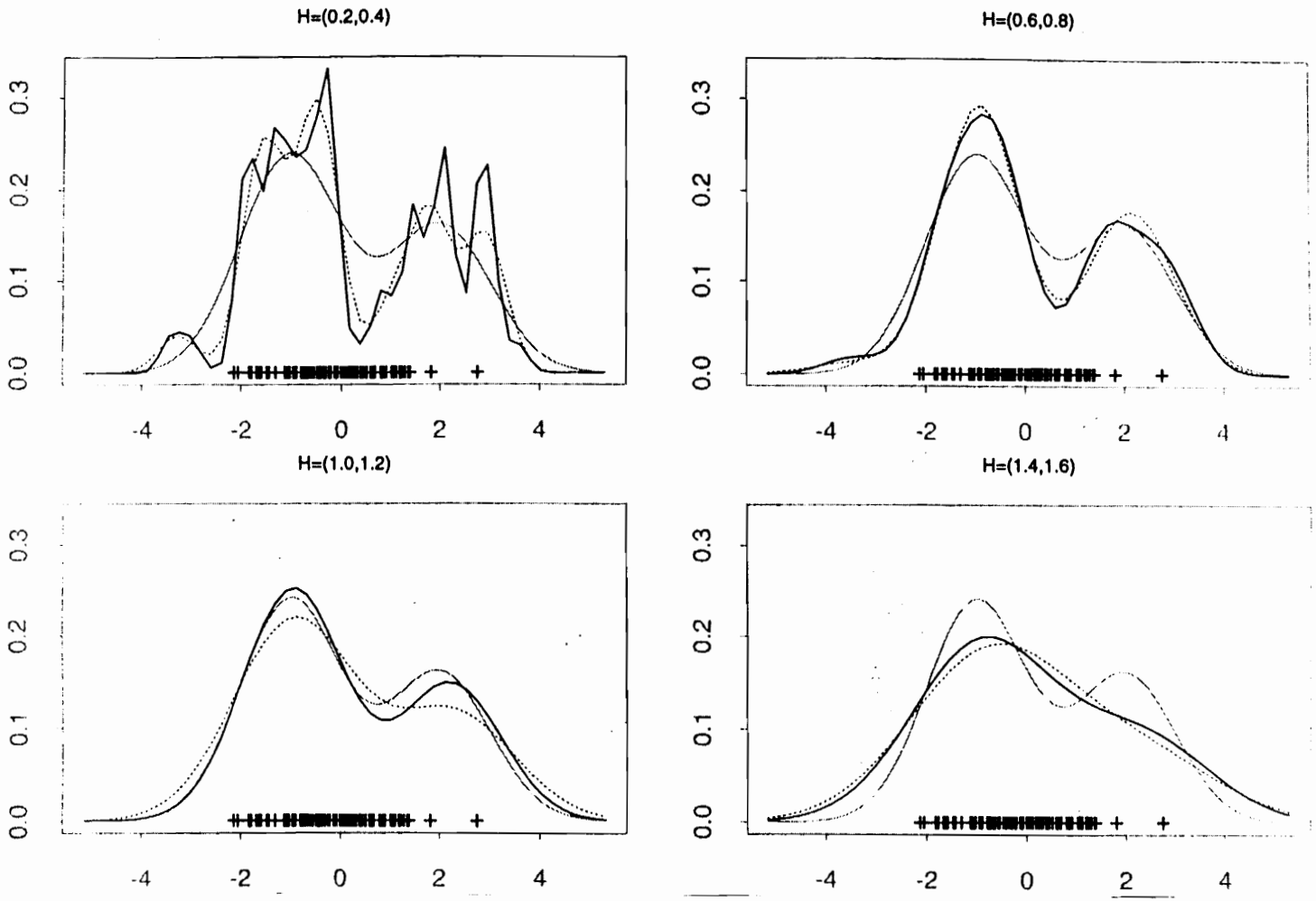


Figure 2-2-3 : DLSMDE for 0.6 $N(-1,1)$ + 0.4 $N(2,1)$ (n=100 with 50 mesh points).The solid line corresponds to the smaller bandwidth. The dotted line corresponds to the larger bandwidth.

Table 2-2-4 : Positive weights for the sample data set by DLSMDE
Data : Random sample of size n=50 from 0.6 N(-1,1) + 0.4 N(2,1)
Optimized with 25 regular mesh points

ID	Mesh pt	H=0.2	H=0.4	H=0.6	H=0.8	H=1.0	H=1.2	H=1.4	H=1.6
1	-3.56546	0.005085	0	0.016023	0.022805	0	0	0	0
2	-3.26257	0.019453	0.040332	0.012916	0	0	0	0	0
3	-2.95967	0.013011	0	0	0	0	0	0	0
6	-2.05098	0.033045	0	0	0	0	0	0	0
7	-1.74809	0.099416	0.08221	0.011137	0	0	0	0	0
8	-1.44519	0.042975	0.181295	0.246041	0	0	0	0	0
9	-1.1423	0.094582	0	0	0.18419	0.228042	0.190261	0	0
10	-0.8394	0.079649	0	0	0.411081	0.402839	0.461835	0.685203	0.62371
11	-0.53651	0.042814	0.213106	0.323527	0	0	0	0	0.086241
12	-0.23361	0.155886	0.091378	0	0	0	0	0	0
14	0.372181	0.004932	0	0	0	0	0	0	0
15	0.675076	0.025637	0	0	0	0	0	0	0
16	0.977971	0.024672	0.03781	0	0	0	0	0	0
17	1.280867	0.033994	0.043034	0	0	0	0	0	0
18	1.583762	0.063618	0.025016	0.194549	0	0	0	0	0
19	1.886658	0.032962	0.132661	0.02469	0.22027	0	0	0	0
20	2.189553	0.083064	0	0	0	0.369118	0.270466	0.167662	0.290049
21	2.492449	0	0	0	0.161654	0	0.077438	0.147135	0
22	2.795344	0.105311	0.111272	0.171117	0	0	0	0	0
23	3.098239	0.026225	0.041887	0	0	0	0	0	0
25	3.70403	0.013668	0	0	0	0	0	0	0
#(positive weights)		20	11	8	5	3	4	3	3
Minimized Q[f,a]		-0.20765	-0.19217	-0.18502	-0.18229	-0.17705	-0.16794	-0.16025	-0.15474

Table 2-2-5 : Number of positive weights for the sample data set by LSMDE
Data : Random sample of size n=50 from 0.6 N(-1,1) + 0.4 N(2,1)

h	LSMDE	DLSMDE(50)	DLSMDE(100)	DLSMDE(25)
0.2	20	18	18	20
0.4	10	11	9	11
0.6	7	7	8	8
0.8	5	7	6	5
1.0	3	3	4	3
1.2	3	4	4	4
1.4	4	3	3	3
1.6	3	3	3	3

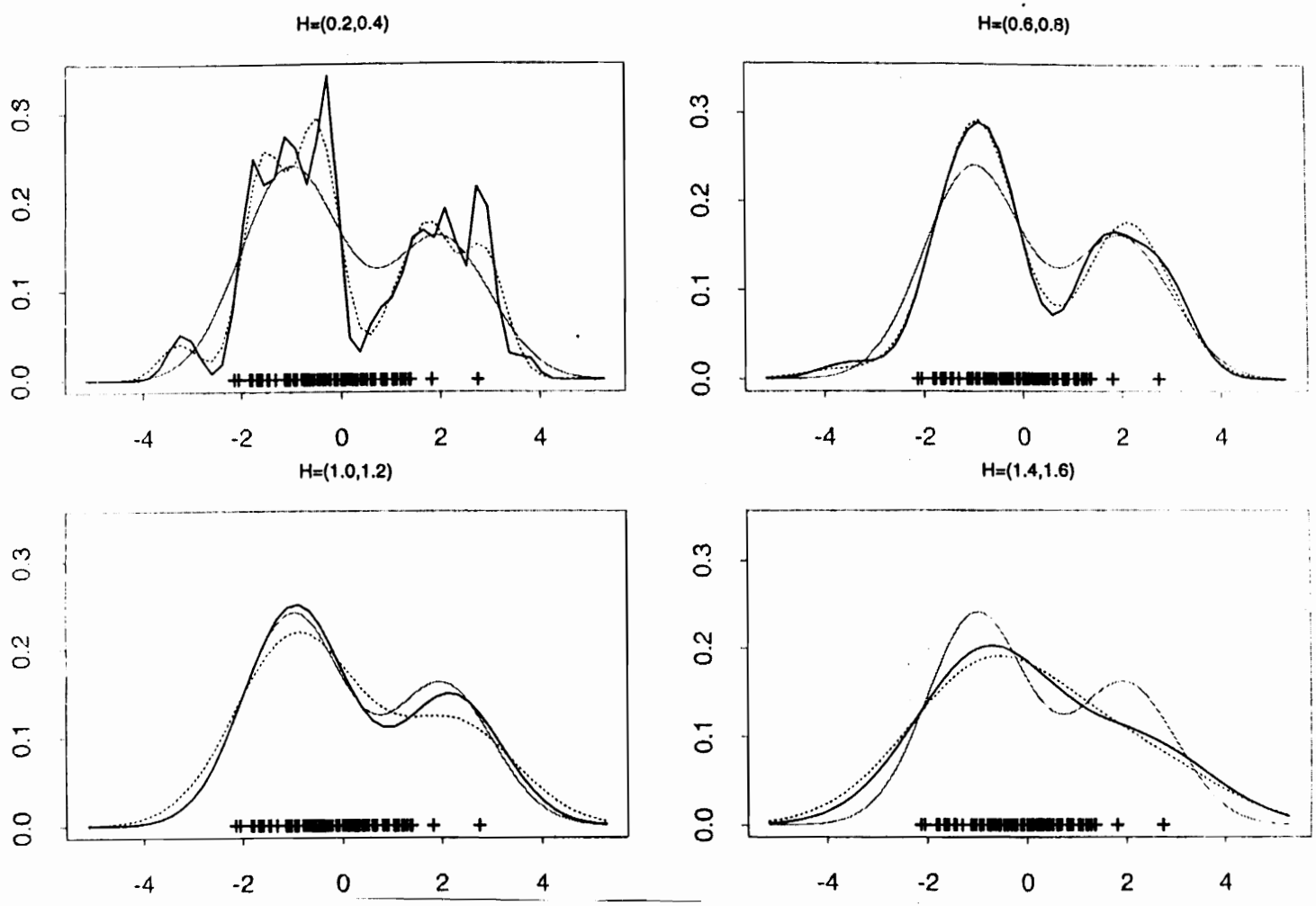


Figure 2-2-4 : DLSMDE for 0.6 $N(-1,1)$ + 0.4 $N(2,1)$ (n=100 with 25 mesh points).The solid line corresponds to the smaller bandwidth. The dotted line corresponds to the larger bandwidth.

2.3.3 Buffalo snow fall data

Let us consider the Buffalo snowfall data to see how the LSMDE works for real data sets. The annual snowfall during 63 winters was recorded from 1910/11 to 1972/73. Originally Parzen examined this data set and suggested that the evidence leans towards a unimodal density while others have argued that the data appear to be trimodal [Silverman (1986) p. 44]. Silverman gives two examples of the kernel density estimates for this data set. For $h = 12$, the kernel density estimates appear to be unimodal and have a roughly normal distribution. However, for $h = 6$, the kernel estimates appear to be trimodal suggesting a mixture of three populations approximately in the ratio 1:3:1 [Silverman (1986) p. 44].

Smoothing parameters from 2.5 to 20.0 by the increment of 2.5 are chosen (Table 2-3-1 with Fig 2-3-1 (A, B)). For $h = 2.5$, the LSMDE looks rough showing many bumps. For $h = 5.0$, the estimate rapidly shows three modes even though the positive weights spread out along the range of data set.

For $h = 7.5$ and 10.0, the estimate still suggests trimodality of the snowfall data. Especially, for $h = 10.0$, we have three conspicuous positive weights clusters. This suggests that the left mode locates around $X_8 = 51.1$ with the weight $a_8 = 0.2449962$, the middle mode between $X_{32} = 79.6$ and $X_{33} = 80.7$ with the weights $a_{32} = 0.1047843$ and $a_{33} = 0.4107039$ and the right mode locates around $X_{57} = 113.7$ with the weight $a_{57} = 0.1221553$. The LSMDE suggests the snowfall distribution can be decomposed into roughly 1:2:1 mixture of normal kernel functions for $h = 10$.

For $h = 12.5$ and above, the estimates become unimodal. However, the estimate still suggests a mixture of three component densities by showing three prominent clusters of positive weights as h increases.

As in previous examples, the discretized modification with mesh points of 100, 50, and 25 gives almost the same result as the ordinary LSMDE. (Table 2-3-2 to Table 2-3-4 with Fig 2-3-2 to Fig 2-3-4).

Table 2-3-1 : Positive weights for the Buffalo snowfall data by LSMDE

Data : Buffalo Snowfall data n=63

ID	Snowfall	H=2.5	H=5.0	H=7.5	H=10.0	H=12.5	H=1.50	H=17.5	H=20.0
1	25	0.016609	0.0134	0.014481	0	0	0	0	0
2	39.8	0	0	0.020916	0	0	0	0	0
3	39.9	0.060467	0.020938	0	0	0	0	0	0
4	40.1	0	0.034293	0	0	0	0	0	0
6	49.1	0	0	0	0	0	0.189732	0.150991	0
7	49.6	0.062681	0	0	0	0.220336	0	0	0
8	51.1	0.013555	0	0	0.244996	0	0	0	0.098792
9	51.6	0	0	0.061583	0	0	0	0	0
10	53.5	0	0.176846	0.160204	0	0	0	0	0
12	55.5	0.082637	0	0	0	0	0	0	0
16	63.6	0.038177	0	0	0	0	0	0	0
17	65.4	0.008509	0	0	0	0	0	0	0
19	69.3	0	0.010193	0	0	0	0	0	0
20	70.9	0	0.141793	0	0	0	0	0	0
22	71.5	0.11982	0	0	0	0	0	0	0
23	71.8	0	0	0.020305	0	0	0	0	0
24	72.9	0	0	0.057855	0	0	0	0	0
26	76.2	0	0	0	0	0	0	0	0.018749
27	77.8	0	0	0	0	0	0	0	0.701105
29	78.4	0	0	0	0	0	0	0.030843	0
30	79	0.130168	0	0	0	0	0	0.628273	0
32	79.6	0	0	0	0.104784	0.249079	0.601263	0	0
33	80.7	0	0.151475	0.227733	0.410704	0.307366	0	0	0
34	82.4	0	0.039638	0.085737	0	0	0	0	0
35	82.4	0	0.039638	0.085737	0	0	0	0	0
36	83	0.059041	0	0	0	0	0	0	0
37	83.6	0.022333	0	0	0	0	0	0	0
38	83.6	0.022333	0	0	0	0	0	0	0
41	87.4	0	0.105947	0	0	0	0	0	0
43	89.6	0.107358	0	0	0	0	0	0	0
48	98.3	0.022784	0	0	0	0	0	0	0
50	102.4	0.020276	0.045155	0	0	0	0	0	0
51	103.9	0.062745	0.074823	0	0	0	0	0	0
53	105.2	0	0	0.11679	0	0	0	0	0
54	110	0	0	0.069886	0	0	0	0	0.181355
55	110.5	0.018013	0	0	0.058681	0	0	0.016426	0
56	110.5	0.018013	0	0	0.05868	0	0	0.016426	0
57	113.7	0.05344	0.091578	0	0.122155	0.223218	0.209005	0.15704	0
61	120.7	0.029576	0.006632	0.078774	0	0	0	0	0
62	124.7	0.019467	0.047652	0	0	0	0	0	0
63	126.4	0.011998	0	0	0	0	0	0	0
#(positive weights)		22	15	12	6	4	3	6	4
Minimized Q[f,a]		-0.20765	-0.19217	-0.18502	-0.18229	-0.17705	-0.16794	-0.16025	-0.15474

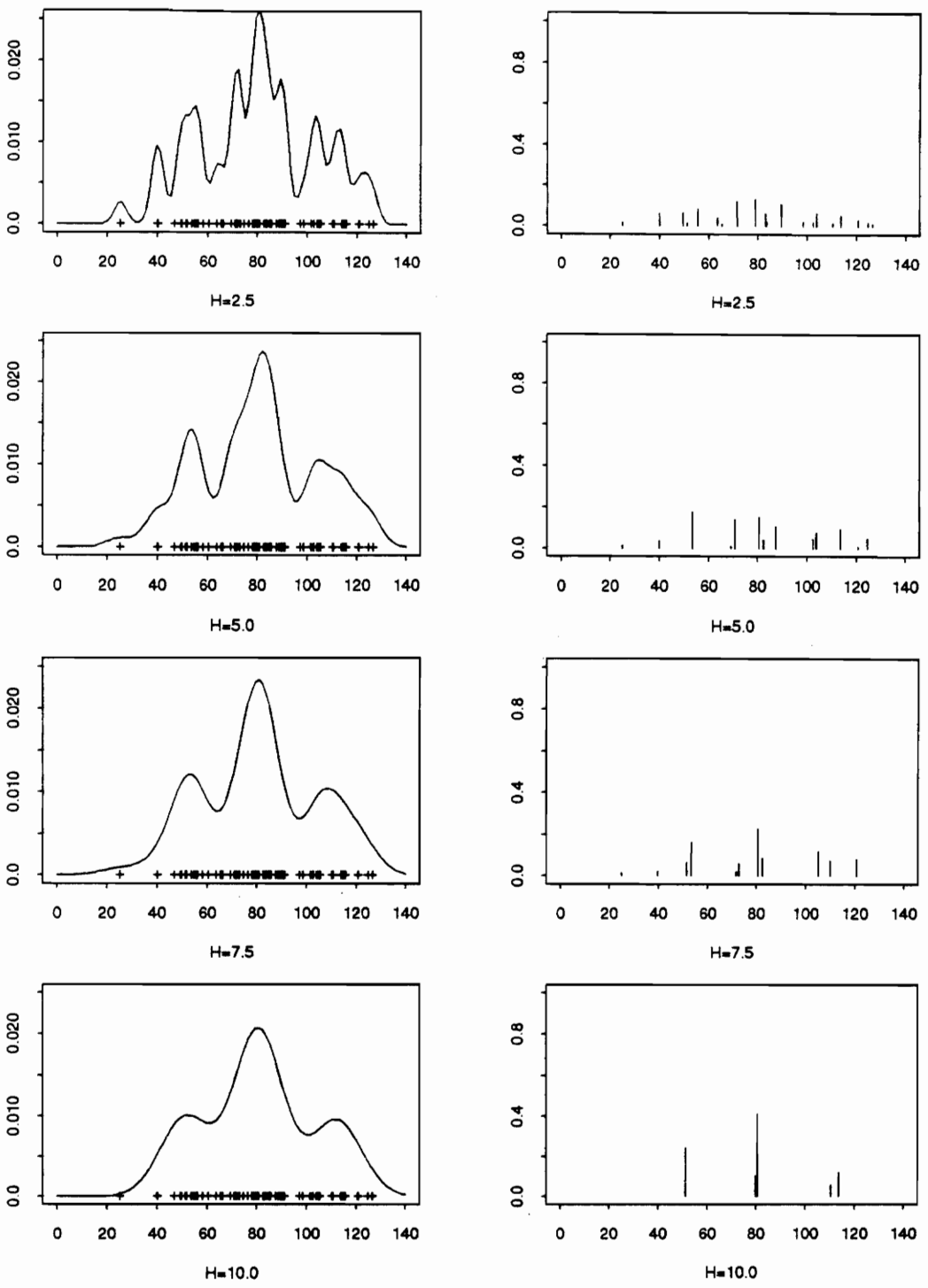


Figure 2-3-1 (A) : LSMDE for Buffalo snowfall data with positive weights (n=63).

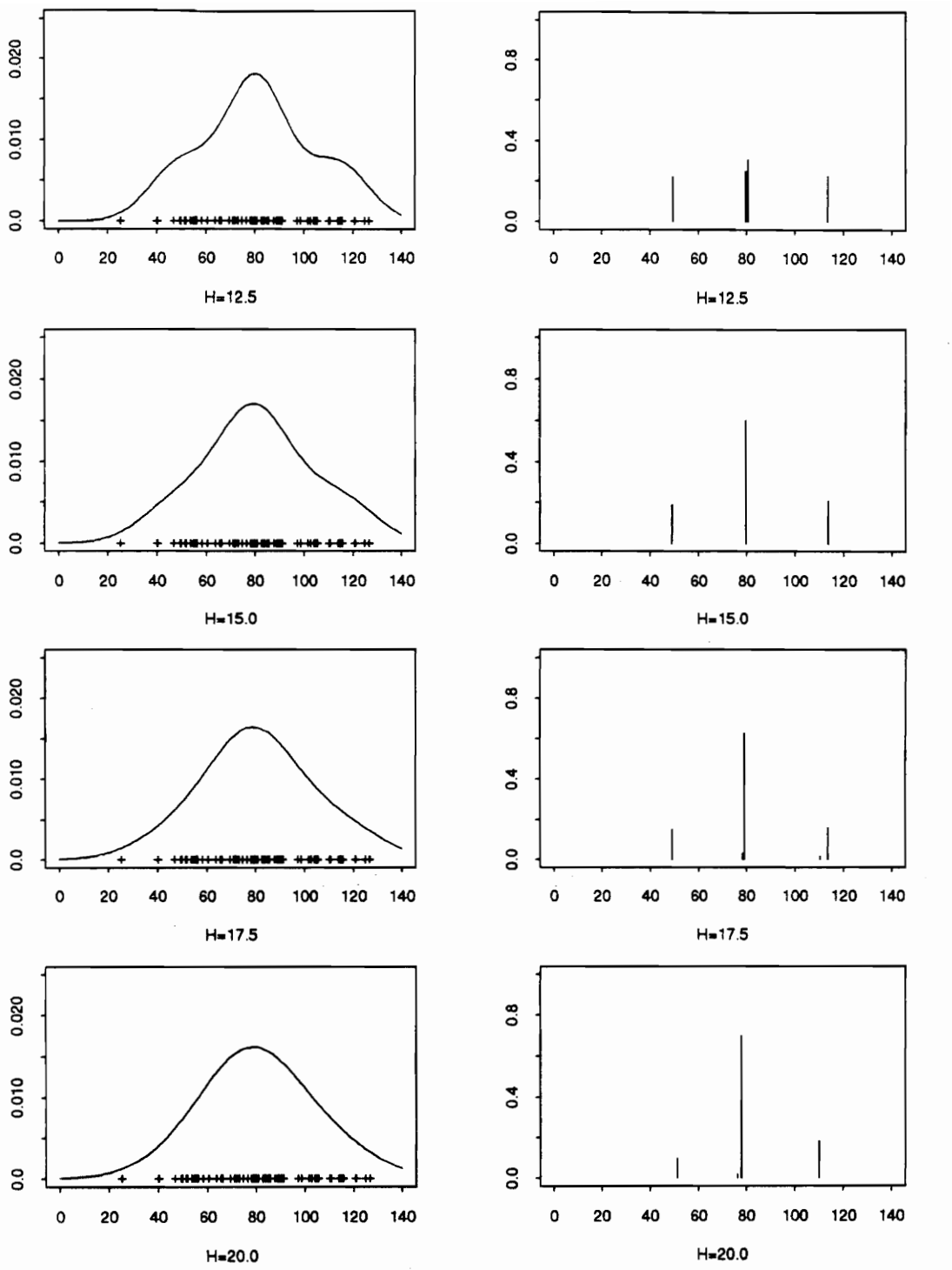


Figure 2-3-1 (B) : LSMDE for Buffalo snowfall data with positive weights (n=63).

Table 2-3-2 : Positive weights for the Buffalo snowfall data by DLSMDE

Data : Buffalo Snowfall data n=63 optimized with 100 mesh points

ID	Mesh pt	H=2.5	H=5.0	H=7.5	H=10.0	H=12.5	H=1.50	H=17.5	H=20.0
1	23.125	0	0	0	0.000727	0	0	0	0
2	24.1871	0	0.014256	0.012175	0	0	0	0	0
3	25.2492	0.016868	0	0	0	0	0	0	0
15	37.9947	0	0	0.02151	0	0	0	0	0
17	40.1189	0.060641	0.055044	0	0	0	0	0	0
25	48.6159	0	0	0	0	0	0.186399	0.145203	0
26	49.678	0.059881	0	0	0	0.221128	0	0	0.08748
27	50.7401	0.015648	0	0	0.156821	0	0	0	0
28	51.8023	0	0	0	0.088293	0	0	0	0
29	52.8644	0	0.074905	0.221759	0	0	0	0	0
30	53.9265	0	0.102101	0	0	0	0	0	0
31	54.9886	0.040394	0	0	0	0	0	0	0
32	56.0508	0.044562	0	0	0	0	0	0	0
39	63.4856	0.001387	0	0	0	0	0	0	0
40	64.5477	0.046231	0	0	0	0	0	0	0
45	69.8583	0	0.02097	0	0	0	0	0	0
46	70.9205	0.032564	0.131336	0	0	0	0	0	0
47	71.9826	0.088644	0	0.063294	0	0	0	0	0
48	73.0447	0	0	0.018945	0	0	0	0	0
52	77.2932	0	0	0	0	0	0	0	0.723746
53	78.3553	0	0	0	0	0	0	0.510879	0
54	79.4174	0.143469	0	0	0	0.141446	0.602657	0.147482	0
55	80.4795	0	0.081171	0	0.515797	0.413497	0	0	0
56	81.5417	0	0.138294	0.398455	0	0	0	0	0
58	83.6659	0.085056	0	0	0	0	0	0	0
61	86.8523	0	0.097441	0	0	0	0	0	0
62	87.9144	0	0.018498	0	0	0	0	0	0
63	88.9765	0.060351	0	0	0	0	0	0	0
64	90.0386	0.047322	0	0	0	0	0	0	0
72	98.5356	0.024315	0	0	0	0	0	0	0
76	102.7841	0.018605	0.081848	0	0	0	0	0	0
77	103.8462	0.064425	0.035561	0	0	0	0	0	0
79	105.9704	0	0	0.010343	0	0	0	0	0
80	107.0326	0	0	0.171232	0	0	0	0	0
82	109.1568	0	0	0	0	0	0	0	0.1387
83	110.2189	0	0	0	0	0	0	0	0.050074
84	111.2811	0.0346	0	0	0.038702	0	0	0	0
85	112.3432	0.025322	0.008267	0	0.19966	0	0	0.196436	0
86	113.4053	0.012889	0.083534	0	0	0.22393	0.210944	0	0
87	114.4674	0.014163	0	0	0	0	0	0	0
93	120.8401	0.033117	0	0.082288	0	0	0	0	0
95	122.9644	0	0.021521	0	0	0	0	0	0
96	124.0265	0	0.035253	0	0	0	0	0	0
97	125.0886	0.015436	0	0	0	0	0	0	0
98	126.1508	0.014109	0	0	0	0	0	0	0
#(positive weights)		24	26	9	6	4	3	4	4
Minimized Q[f,a]		-0.01465	-0.01319	-0.01287	-0.01261	-0.01212	-0.01179	-0.01162	-0.01156

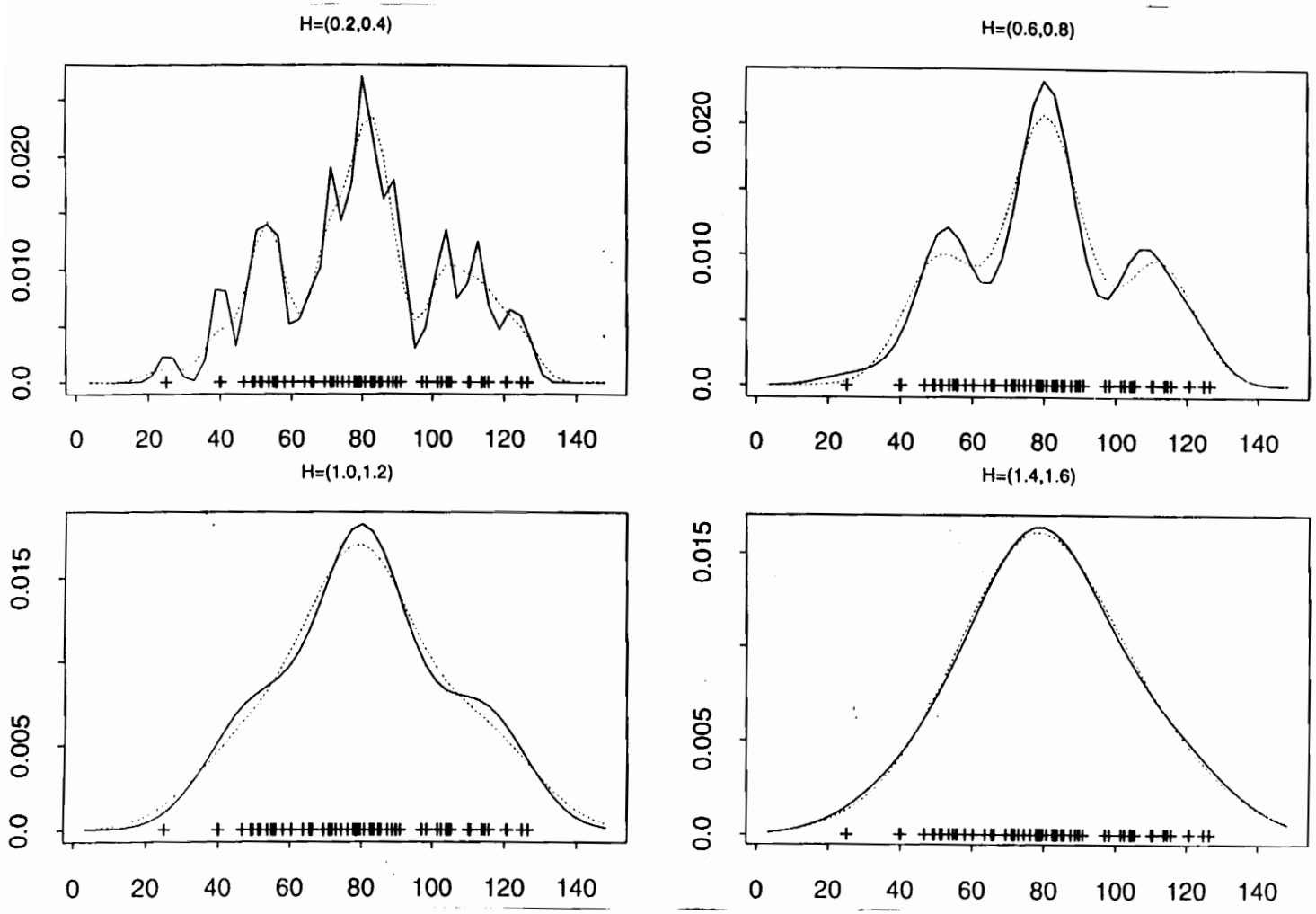


Figure 2-3-2 : DLSMDE for Buffalo snowfall data (n=63 with 100 mesh points).The solid line corresponds to the smaller bandwidth. The dotted line corresponds to the larger bandwidth.

Table 2-3-3 : Positive weights for the Buffalo snowfall data by DLSMDE

Data : Buffalo Snowfall data n=63 optimized with 50 mesh points

ID	Mesh pt	H=2.5	H=5.0	H=7.5	H=10.0	H=12.5	H=1.50	H=17.5	H=20.0
1	23.125	0	0.014143	0.005545	0.000276	0	0	0	0
2	25.2709	0.017092	0	0.006619	0	0	0	0	0
8	38.1464	0	0	0.022378	0	0	0	0	0
9	40.2923	0.059933	0.053487	0	0	0	0	0	0
13	48.876	0.036558	0	0	0	0.161579	0.187015	0.149562	0.081135
14	51.0219	0.040134	0	0.019326	0.244553	0.057735	0	0	0
15	53.1679	0	0.169382	0.20348	0	0	0	0	0
16	55.3138	0.079622	0.00608	0	0	0	0	0	0
17	57.4597	0.001492	0	0	0	0	0	0	0
20	63.8974	0.045046	0	0	0	0	0	0	0
23	70.3352	0.05704	0.14803	0	0	0	0	0	0
24	72.4811	0.066072	0.000343	0.077244	0	0	0	0	0
26	76.773	0	0	0	0	0	0	0	0.649558
27	78.9189	0.125613	0	0	0.135577	0.24897	0.462973	0.660199	0.075173
28	81.0648	0	0.234281	0.327276	0.381456	0.307346	0.13847	0	0
29	83.2107	0.108121	0	0.074359	0	0	0	0	0
31	87.5026	0	0.108315	0	0	0	0	0	0
32	89.6485	0.108009	0	0	0	0	0	0	0
36	98.2321	0.01751	0	0	0	0	0	0	0
38	102.524	0.057009	0.086645	0	0	0	0	0	0
39	104.6699	0.031028	0.029596	0	0	0	0	0	0
40	106.8158	0	0	0.178034	0	0	0	0	0
41	108.9617	0	0	0	0	0	0	0	0.194135
42	111.1077	0.036241	0	0	0.120272	0	0	0.014894	0
43	113.2536	0.050184	0.093386	0	0.117866	0.22437	0.211542	0.175346	0
46	119.6913	0.01051	0	0.056805	0	0	0	0	0
47	121.8372	0.028367	0.001365	0.028935	0	0	0	0	0
48	123.9832	0	0.054896	0	0	0	0	0	0
49	126.1291	0.02442	0	0	0	0	0	0	0
#(positive weights)		20	13	11	6	5	4	4	4
Minimized Q[f,a]		-0.01456	-0.01319	-0.01288	-0.0126	-0.01212	-0.01178	-0.01162	-0.01156

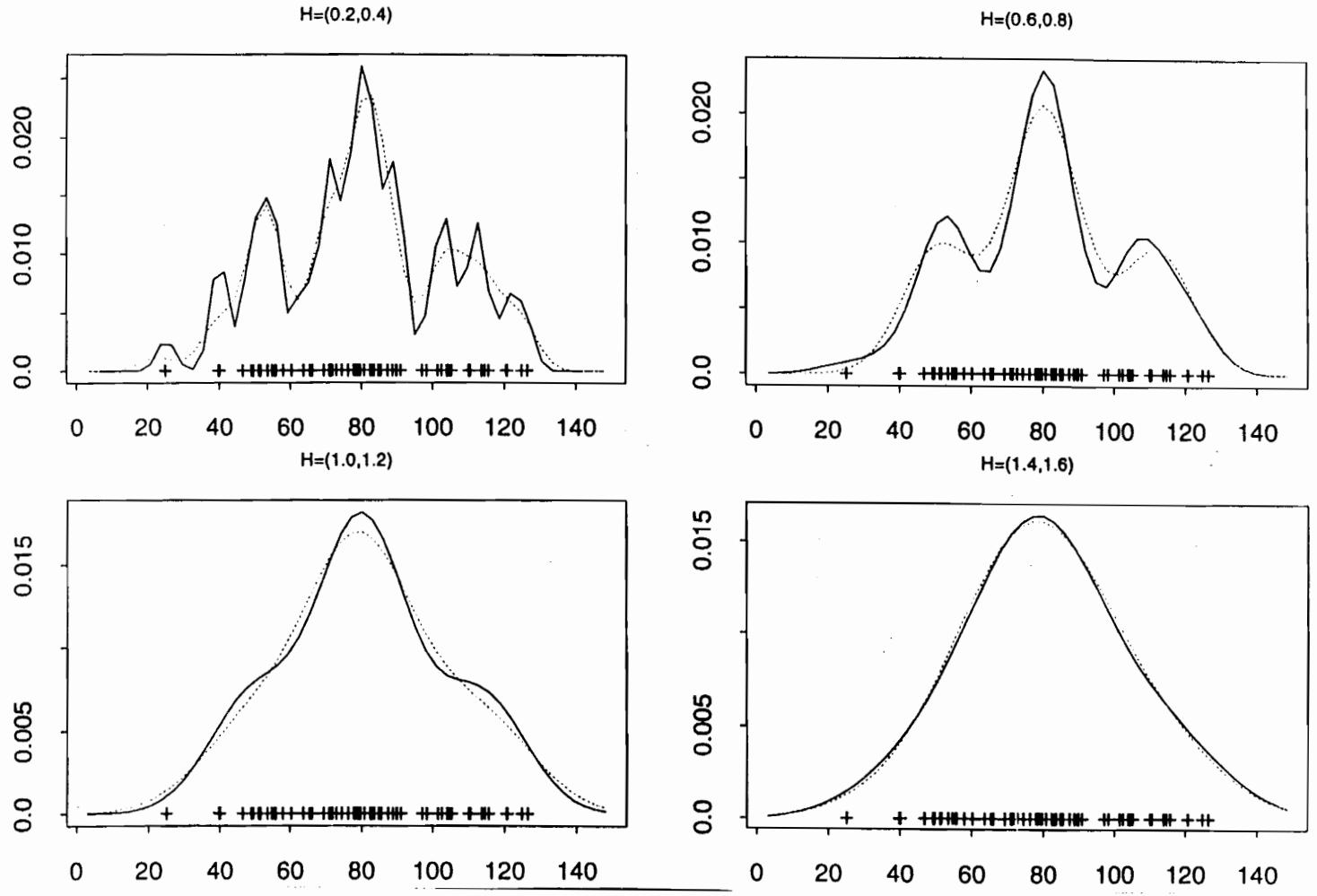


Figure 2-3-3 : DLSMDE for Buffalo snowfall data (n=63 with 50 mesh points).The solid line corresponds to the smaller bandwidth. The dotted line corresponds to the larger bandwidth.

Table 2-3-4 : Positive weights for the Buffalo snowfall data by DLSMDE

Data : Buffalo Snowfall data n=63 optimized with 25 mesh points

ID	Mesh pt	H=2.5	H=5.0	H=7.5	H=10.0	H=12.5	H=1.50	H=17.5	H=20.0
1	23.125	0.012104	0.014735	0.010899	0	0	0	0	0
2	27.5062	0.005717	0	0	0	0	0	0	0
4	36.2687	0	0	0.017414	0	0	0	0	0
5	40.65	0.060934	0.058189	0.004304	0	0	0	0	0
7	49.4125	0.057578	0	0.040725	0.172224	0.218844	0.195451	0.152385	0.086051
8	53.7937	0.079739	0.176023	0.187319	0.070908	0	0	0	0
9	58.175	0.021895	0	0	0	0	0	0	0
10	62.5562	0.038375	0	0	0	0	0	0	0
11	66.9375	0.002842	0.000801	0	0	0	0	0	0
12	71.3187	0.129436	0.155508	0.04355	0	0	0	0	0
13	75.7	0.001999	0	0	0	0	0	0.157966	0.415415
14	80.0812	0.172606	0.118967	0.342898	0.51432	0.555735	0.60309	0.500017	0.323012
15	84.4625	0.043903	0.181458	0.089229	0	0	0	0	0
16	88.8437	0.112659	0.023857	0	0	0	0	0	0
18	97.6062	0.011875	0	0	0	0	0	0	0
19	101.9875	0.07987	0.103134	0	0	0	0	0	0
20	106.3687	0.009436	0	0.170522	0	0	0	0	0
21	110.75	0.062952	0.069345	0	0.187348	0.079997	0.018653	0.104077	0.175522
22	115.1312	0.038216	0.042351	0	0.0552	0.145425	0.182807	0.085555	0
23	119.5125	0.012238	0	0.092746	0	0	0	0	0
24	123.8937	0.042903	0.055632	0.000395	0	0	0	0	0
25	128.275	0.002724	0	0	0	0	0	0	0
#(positive weight)		21	12	11	5	4	4	5	4
Minimized Q[f,a]		-0.0142	-0.01316	-0.01287	-0.01259	-0.01212	-0.01178	-0.01162	-0.01156

Table 2-3-5 : Number of positive weights for Buffalo snowfall data by LSMDE

Data : Buffalo Snowfall data n=63

h	LSMDE	DLSMDE(50)	DLSMDE(100)	DLSMDE(25)
0.2	**20**	**18**	**18**	**20**
0.4	**10**	**11**	**9**	**11**
0.6	**7**	**7**	**8**	**8**
0.8	**5**	**7**	**6**	**5**
1.0	**3**	**3**	**4**	**3**
1.2	**3**	**4**	**4**	**4**
1.4	**4**	**3**	**3**	**3**
1.6	**3**	**3**	**3**	**3**

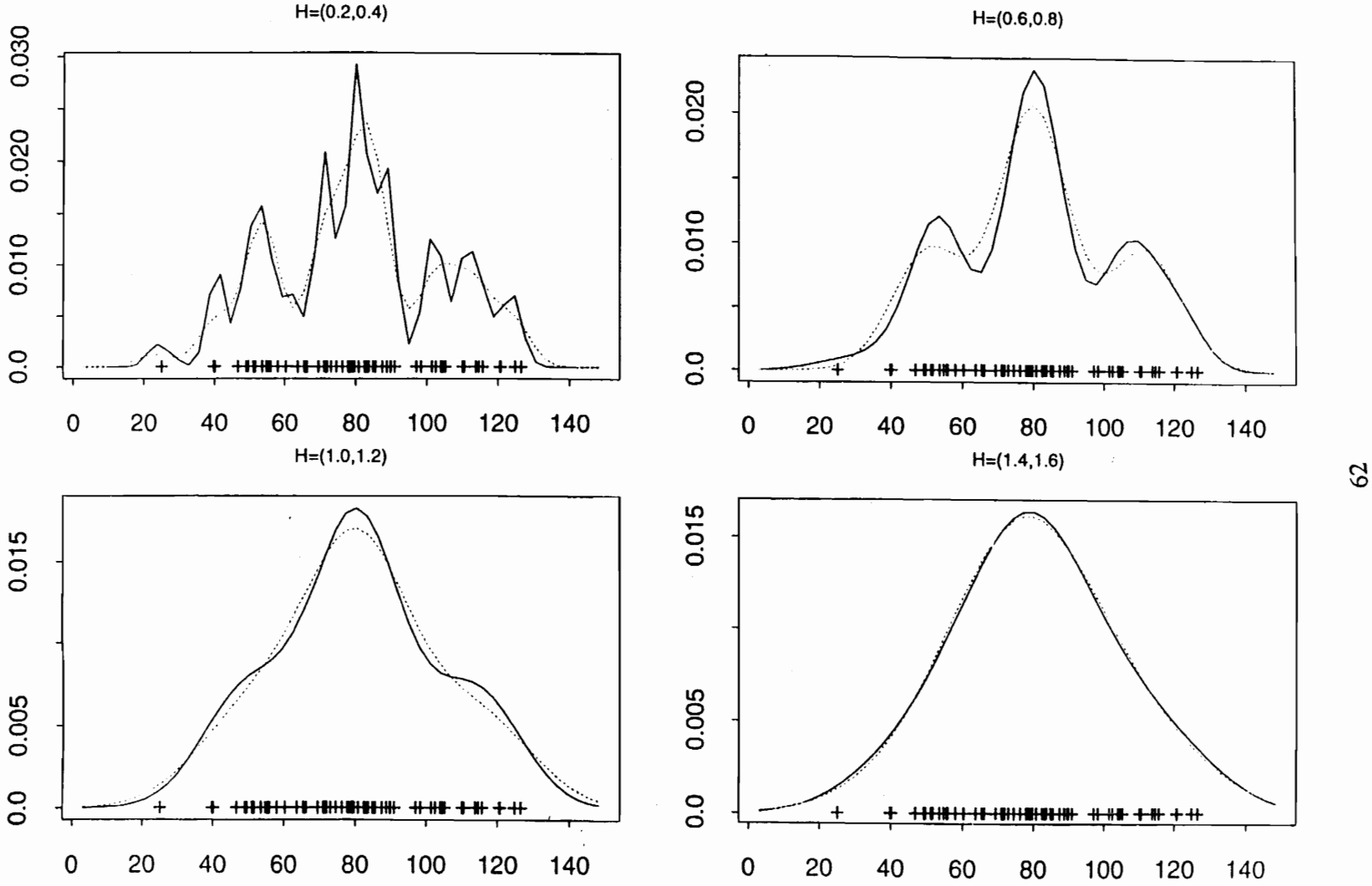


Figure 2-3-4 : DLSMDE for Buffalo snowfall data (n=63 with 25 mesh points).The solid line corresponds to the smaller bandwidth. The dotted line corresponds to the larger bandwidth.

2.4 Application to exponential mixtures

The exponential distribution and its modifications such as Weibull distribution have been widely employed as a model in life time distributions. When one considers that failures may arise for a number of different sources, a mixture of those distributions might provide a better description of the failure properties by imposing several exponential distributions, each representing a type of failure. In this section, we illustrate how the LSMDE can be applied in the identification of exponential mixture models without the problem of the optimal smoothing parameter. Since there is no burden of determining the optimal smoothing parameter, the LSMDE applied to the case of exponential mixtures can be considered as a mixture estimation technique as well as a nonparametric density estimator. Other available methods for finite mixtures such as the method of moments, and the maximum likelihood (the EM algorithm for mixture) will be briefly compared to the LSMDE.

2.4.1 Mixture of exponential distributions and its transformation

The general form of a finite exponential mixture is

$$f_U(u) = \sum_{j=1}^{k} \pi_j \frac{1}{\mu_j} \exp(-\frac{u}{\mu_j}) \quad \text{if } u \geq 0 \tag{2.30}$$

$$= 0 \text{ otherwise}$$

where $\pi_j \geq 0$, $\mu_j > 0$ for all $j = 1,2,\ldots,k$ and $\sum_{j=1}^{k} \pi_j = 1$.

The scale mixture (2.30) can be transformed to a location mixture by letting $x = \ln(u)$, and $\alpha_j = \ln(\mu_j)$ for all $j = 1,2,...,k$. Then X has a probability density function of

$$f_X(x) = \sum_{j=1}^{k} \pi_j K(x - \alpha_j) = \sum_{j=1}^{k} \pi_j \exp(x - \alpha_j)\exp(-\exp(x - \alpha_j)) \tag{2.31}$$

where $K(u) = K(x - \alpha_j)$ is called the Gumbel density, or the probability density function of least extreme value with location parameter, $\alpha = \alpha_j$ and scale parameter, $\beta = 1$.

We define the LSMDE with Gumbel kernels as follows. $\hat{f}_X(y)$ is defined as the LSMDE of (2.31) if given an i.i.d. random sample of size n , $\{X_i\}$, $i = 1,2,...n$, the kernel function , $K(u) = K_{h=1}(u) = \exp(u)\exp(-\exp(u))$,

$$\hat{f}_X(y) = \sum_{i=1}^{n} a_i K_h(y - x_i) = \sum_{i=1}^{n} a_i \exp(y - x_i)\exp(-\exp(y - x_i))$$

where $\{a_i\}$, $i = 1,2,...n$ minimize the objective function,

$$\hat{Q}[\hat{f}_X(y)] = -\frac{2}{n}\sum_{i=1}^{n} \hat{f}_X(X_i) + \int \hat{f}_X{}^2(y)dy = \underline{a}^T \underline{d} + \frac{1}{2}\underline{a}^T C \underline{a}$$

$$\text{subject to } \sum_{i=1}^{n} a_i = 1 \text{ and } a_i \geq 0 \text{ for all } i. \tag{2.32}$$

where $$\underline{d} = -\frac{2}{n}\left(\sum_{j=1}^{n} K(x_1 - x_j), \sum_{j=1}^{n} K(x_2 - x_j), \cdots, \sum_{j=1}^{n} K(x_n - x_j)\right)^T$$

and the matrix of convolution,

$$C = 2\begin{bmatrix} K^*(0) & K^*(x_1 - x_2) & \cdots & K^*(x_1 - x_n) \\ K^*(x_2 - x_1) & K^*(0) & \cdots & K^*(x_1 - x_2) \\ \vdots & \vdots & \ddots & \vdots \\ K^*(x_n - x_1) & K^*(x_n - x_2) & \cdots & K^*(0) \end{bmatrix}.$$

It can be shown that $K^*(x_i - x_j) = \exp(x_i - x_j)/(1+\exp(x_i - x_j))^2$ is the probability density function of the standard logistic distribution (see Appendix C for the details). Although $K(u) = K_{h=1}(u) = \exp(u)\exp(-\exp(u))$, the Gumbel kernel is not symmetric about 0 – negatively skewed, its convolution $K^*(u) = \exp(u)/(1+\exp(u))^2$, the standard logistic distribution is symmetric about zero. Hence the LSMDE (2.30) satisfies all the properties such as existence and uniqueness derived from the previous chapter.

After solving the quadratic programming problem (2.30) for a_i's, we expect a set of positive weights reflects mixing proportions and x_i's corresponding to positive weights reflect location parameters, $\alpha_j = \ln(\mu_j)$ in the Gumbel mixture (2.31). Since (2.32) does not contain the smoothing parameter h, the LSMDE in estimation of exponential mixtures does not have the problem of optimal bandwidth selection.

2.4.2. Method of moments and the EM algorithm for mixtures

For mixtures of the normal distributions, the comparison of the LSMDE with other available methods for mixture problems is not convenient due to the determination of the optimal smoothing parameter. Since the LSMDE for exponential mixtures is independent of the smoothing parameter, comparing the LSMDE with other available estimators for the finite mixture case will be helpful to understand its performance. Here,

we review two methods, the method of moments by Rider (1961) and the estimator by the EM algorithm [Titterington, Smith and Makov (1985)] for exponential mixtures. Before we proceed, we emphasize that the number of components in (2.31) is assumed known – 2 for the method of moments, k for the EM algorithm.

Rider (1961) has applied the method of moments to the decomposition of mixtures of two exponential distributions. Let m_i be the i-th moment about zero and $T_i = \frac{m_i}{\Gamma(i+1)}$. Then $(\hat{\pi}, \hat{\mu}_1, \hat{\mu}_2)$, the estimators by the methods of moments are obtained from the systems of equations,

$$\begin{aligned} \pi\,\mu_1 + (1-\pi)\,\mu_2 &= T_1, \\ \pi\,\mu_1^2 + (1-\pi)\,\mu_2^2 &= T_2, \\ \pi\,\mu_1^3 + (1-\pi)\,\mu_2^3 &= T_3. \end{aligned} \tag{2.33}$$

$\hat{\mu}_j$ can be obtained from the quadratic equation,

$$6(2m_1^2 - m_2)\hat{\mu}_j^2 + 2(m_3 - 3m_1m_2)\hat{\mu}_j + 3m_2^2 - 2m_1m_3 = 0 \tag{2.34}$$

and $\hat{\pi} = \frac{m_1 - \hat{\mu}_2}{\hat{\mu}_1 - \hat{\mu}_2}$. Although it is possible that the roots of (2.34) will not both be positive, or even real, the estimators are known to be consistent provided that $\hat{\mu}_1 \neq \hat{\mu}_2$. The fact that $\hat{\mu}_1$ and $\hat{\mu}_2$ are not both positive unless the sequence

$$\frac{T_0}{T_1}, \quad \frac{T_1}{T_2}, \quad \frac{T_2}{T_3}$$

is monotonic (increasing or decreasing) permits a quick check on whether the method will work [Everitt and Hand (1981)].

Unlike simple parametric models, the maximum likelihood approach for mixture models is not always so straightforward. Explicit solution for the likelihood equation may not be possible, and we need to use iterative computation of the solution, such as Newton-Raphson, the method of scoring or the EM algorithm. The EM algorithm developed by Dempster, Laird, and Rubin (1977) has been widely used to approximate maximum likelihood estimates for incomplete data problems. The formulation and theoretical properties of the EM algorithm for mixture densities, especially from exponential families has been discussed in Redner and Walker (1984). We offer a brief review of the EM algorithm for exponential mixtures based on the discussion of Titterington, Smith and Makov (1985). For notational convenience, we replace scale parameter μ_j by its inverse λ_j in (2.30).

Since the original application of the EM algorithm is for incomplete data problems, we first interpret mixture problems as incomplete data problems. Suppose $\{x_i\}_{i=1}^n$ be the observations from a finite mixture density with k components and f_{ij} denote $f_j(x_i|\theta_j)$. Define $\{y_i\}_{i=1}^n = \{(x_i, \underline{z}_i)\}_{i=1}^n$ where $\underline{z}_i = (z_{i1}, z_{i2}, \ldots, z_{ik})^T$ and $z_{ij} = 1$ if x_i belongs to the j-th component, $z_{ij} = 0$, otherwise. Then the likelihood function and log-likelihood function of $(y_1, y_2, \ldots, y_n)$ are written in the form

$$g(y_1, \ldots, y_n|\underline{\psi}) = \prod_{i=1}^{n}\prod_{j=1}^{k}(\pi_j f_{ij})^{z_{ij}},$$

$$\log g(y_1, \ldots, y_n|\underline{\psi}) = \sum_{i=1}^{n}\sum_{j=1}^{k}(z_{ij}\log\pi_j + z_{ij}\log f_{ij}) = \sum_{i=1}^{n}\underline{z}_i^T\,\underline{v}(\underline{\pi}) + \sum_{i=1}^{n}\underline{z}_i^T\,\underline{u}_i(\underline{\theta}).$$

where $\underline{z}_i = (z_{i1},\ldots,z_{ik})^T$, $\underline{v}(\underline{\pi}) = (\log\pi_1,\ldots,\log\pi_k)^T$, $\underline{u}_i(\underline{\theta}) = (\log f_{i1},\ldots\log f_{ik})^T$, and $\underline{\psi} = (\underline{\pi},\underline{\theta}) = (\pi_1,\ldots,\pi_k;\theta_1,\ldots,\theta_k)^T$. Here, $\underline{z}_1,\ldots,\underline{z}_n$ are considered as missing observations.

The EM algorithm consists of iteration of 2 steps – Expectation Step (E-step) and Maximization Step (M-step). Given a current approximation $\underline{\psi}^{(m)}$and $\underline{x}$, define the conditional expectation of log-likelihood function of complete data, $\underline{y}$,

$$Q[\underline{\psi},\underline{\psi}^{(m)}] = E[\log g(\underline{y}|\underline{\psi})|\underline{x},\underline{\psi}^{(m)}].$$

Each iteration consists of the following two steps:

E-step : Evaluate $Q[\underline{\psi},\underline{\psi}^{(m)}] = E[\log g(\underline{y}|\underline{\psi})|\underline{x},\underline{\psi}^{(m)}]$

M-step : Find $\underline{\psi}^{(m+1)}$ to maximize $Q[\underline{\psi},\underline{\psi}^{(m)}]$.

In many cases such as the exponential family, the actual maximization is explicit.

For the E-step in mixture problems, we have

$$Q[\underline{\psi},\underline{\psi}^{(m)}] = E[\log g(\underline{y}|\underline{\psi})|\underline{x},\underline{\psi}^{(m)}]$$

$$= E[\sum_{i=1}^{n} \underline{z}_i^{\,T}\,\underline{v}(\underline{\pi}) + \sum_{i=1}^{n} \underline{z}_i^{\,T}\,\underline{u}_i(\underline{\theta}) \quad |\,\underline{x},\underline{\psi}^{(m)}]$$

$$= \sum_{i=1}^{n} \underline{w}_i^T(\underline{\psi}^{(m)})\,\underline{v}(\underline{\pi}) \quad + \quad \sum_{i=1}^{n} \underline{w}_i^T(\underline{\psi}^{(m)})\,\underline{u}_i(\underline{\theta}) \qquad (2.35)$$

where $\underline{w}_i(\underline{\psi}^{(m)}) = E[\underline{z}_i|\underline{x},\underline{\psi}^{(m)}] = \left\{E[z_{ij}|\underline{x},\underline{\psi}^{(m)}]\right\}_{j=1}^{k}$. Note $w_{ij}(\underline{\psi}^{(m)}) = E[z_{ij}|\underline{x},\underline{\psi}^{(m)}]$ is the conditional expectation of the binomial distribution with parameters n=1, $p = p^* = \Pr(X_i$ belongs to the j-th component) given $\underline{\psi}^{(m)}$ and $\underline{x}$. More precisely,

$$p^* = \Pr(X_i \text{ belongs to the } j\text{-th component})$$

$$= \frac{\pi_j^{(m)} f_{ij}^{(m)}}{\sum_{j=1}^{k} \pi_j^{(m)} f_{ij}^{(m)}} = \frac{\pi_j^{(m)} f_{ij}^{(m)}}{f(x_i|\underline{\psi}^{(m)})} . \tag{2.36}$$

$w_{ij}(\underline{\psi}^{(m)})$'s are therefore the probabilities of category membership for the i-th observation, conditional on x_i and given that the parameter is $\underline{\psi}^{(m)}$. Since the parameters $\underline{\pi}$ and $\underline{\theta}$ are usually distinct, the M-step can be carried out separately for $\underline{\pi}$ and $\underline{\theta}$. As a consequence, the M-step for $\underline{\pi}$ is

$$\pi_j^{(m+1)} = \frac{1}{n}\sum_{i=1}^{n} w_{ij}(\underline{\psi}^{(m)}) = \frac{n_j(\underline{\psi}^{(m)})}{n}, \quad \text{for } j = 1,\ldots,k \tag{2.37}$$

where $n_j(\underline{\psi}^{(m)})$ corresponds to a pseudo sample size associated with the j-th subpopulation, with observations allocated by the fractions defined by the weights, to the various subpopulations.

Although the form of the M-step for $\underline{\theta}$ specifically depends on the component density, it corresponds, in general, to maximization of the second term in $Q[\underline{\psi},\underline{\psi}^{(m)}]$ (2.35). Furthermore, when the component density belongs to the exponential family, the M-step for $\underline{\theta}$ can be reduced to computing the expected value of the complete-data

sufficient statistic for $\underline{\theta}$. Since the maximization of $Q[\underline{\psi}, \underline{\psi}^{(m)}]$ with respect to $\underline{\theta}$ is reduced to the maximization of $\sum_{i=1}^{n} \underline{w}_i^T(\underline{\psi}^{(m)})\, \underline{u}_i(\underline{\theta})$ in (2.35), we wish to maximize

$$\max_{\underline{\theta}} \quad \sum_{i=1}^{n} \underline{w}_i^T(\underline{\psi}^{(m)}) \begin{pmatrix} \log f_{i1} \\ \vdots \\ \log f_{ik} \end{pmatrix}. \tag{2.38}$$

If the component density belongs to the exponential family, (2.38) becomes

$$\max_{\underline{\theta}} \quad \sum_{i=1}^{n} \underline{w}_i^T(\underline{\psi}^{(m)}) \begin{pmatrix} b(x_i) + t(x_i)\theta_1 - a(\theta_1) \\ \vdots \\ b(x_i) + t(x_i)\theta_k - a(\theta_k) \end{pmatrix}. \tag{2.39}$$

Since $b(x_i)$'s are free of θ_j, $b(x_i)$ can be excluded from the maximization (2.39). Therefore we yield the final function to be maximized for the M-step for $\underline{\theta}$,

$$\max_{\underline{\theta}} \quad \sum_{i=1}^{n} \left(w_{i1}(\underline{\psi}^{(m)}), \ldots, w_{ik}(\underline{\psi}^{(m)}) \right) \begin{pmatrix} t(x_i)\theta_1 - a(\theta_1) \\ \vdots \\ t(x_i)\theta_k - a(\theta_k) \end{pmatrix},$$

or equivalently,

$$\max_{\underline{\theta}} Q^*[\underline{\psi}, \underline{\psi}^*] = \max_{\underline{\theta}} \quad \sum_{i=1}^{n} \sum_{j=1}^{k} w_{ij}(\underline{\psi}^{(m)}) \left[t(x_i)\theta_j - a(\theta_j) \right]. \tag{2.40}$$

The maximization step (2.40) gives the stationary condition,

$$0 = \frac{\partial}{\partial \theta_j} Q^*[\underline{\psi}, \underline{\psi}^{(m)}] = \sum_{i=1}^{n} w_{ij}(\underline{\psi}^{(m)}) [t(x_i) - \phi_j], \; j = 1, \ldots, k \tag{2.41}$$

where $\phi_j = \frac{\partial}{\partial \theta_j} a(\theta_j)$.

By solving (2.41) for ϕ_j, we have

$$\phi_j = \frac{\sum_{i=1}^{n} w_{ij}(\underline{\psi}^{(m)})\, t(x_i)}{\sum_{i=1}^{n} w_{ij}(\underline{\psi}^{(m)})} = \frac{\sum_{i=1}^{n} w_{ij}(\underline{\psi}^{(m)})\, t(x_i)}{n_j(\underline{\psi}^{(m)})} = \frac{\sum_{i=1}^{n} w_{ij}(\underline{\psi}^{(m)})\, t(x_i)}{n\pi_j^{(m+1)}}. \qquad (2.42)$$

Therefore, the M-step for $\underline{\theta}$ can be carried out using the iteration scheme of

$$\phi_j^{(m+1)} = \frac{\sum_{i=1}^{n} w_{ij}(\underline{\psi}^{(m)})\, t(x_i)}{n\pi_j^{(m+1)}}, \quad \text{for } j = 1,\ldots,k, \qquad (2.43)$$

provided that the component density belongs to the exponential family. $\underline{\theta}$ can be obtained from the relationship of $\phi_j = \frac{\partial}{\partial \theta_j} a(\theta_j)$.

Additionally, the approximations generated by the EM algorithm maintains the relationship, in that

$$\sum_{j=1}^{k} \pi_j^{(m)} \phi_j^{(m)} = \frac{1}{n} \sum_{i=1}^{n} t(x_i) \text{ for all } m \text{ except possibly } m=0$$

due to the fact that $\phi_j = \frac{\partial}{\partial \theta_j} a(\theta_j) = E[t(X)|\theta_j]$.

Since the exponential distribution in (2.30) belongs to the one parameter exponential family of distribution of the form $\log f_j(x|\theta_j) = b(x) + t(x)\,\theta_j - a(\theta_j)$

where $b(x)=0$, $t(x)=-x$, $\theta_j=\lambda_j$, and $a(\theta_j)=-\log\lambda_j$, we can summarize the EM algorithm for the finite mixture of exponential distributions as follows.

Preliminary relation :

$$\phi_j=\frac{\partial}{\partial\theta_j}a(\theta_j)=-\frac{1}{\lambda_j},\ \text{for } j=1,\ldots,k.$$

Repeat until converge

M-step for $\underline{\pi}$:

for $j=1,\ldots,k$,

for $i=1,\ldots,n$,

$$w_{ij}(\underline{\psi}^{(m)})=\frac{\pi_j^{(m)}\lambda_j^{(m)}\exp(-\lambda_j^{(m)}x_i)}{\sum_{j=1}^{k}\pi_j^{(m)}\lambda_j^{(m)}\exp(-\lambda_j^{(m)}x_i)}$$

for $j=1,\ldots,k$,

$$\pi_j^{(m+1)}=\frac{1}{n}\sum_{i=1}^{n}w_{ij}(\underline{\psi}^{(m)})$$

M-step for $\underline{\lambda}$:

for $j=1,\ldots,k$,

$$\phi_j^{(m+1)}=\frac{-\sum_{i=1}^{n}w_{ij}(\underline{\psi}^{(m)})\,x_i}{n\pi_j^{(m+1)}}$$

$$\lambda_j^{(m+1)}=-\frac{1}{\phi_j^{(m+1)}}$$

For each iteration, $-\sum_{j=1}^{k}\frac{\pi_j^{(m)}}{\lambda_j^{(m)}}$ should be equal to $-\frac{1}{n}\sum_{i=1}^{n}x_i$ except possibly m=0.

The theoretical results and the problems such as initial estimates and slow convergence are abstracted in Redner and Walker (1984).

2.4.3. Examples and comparison

The methods described in section 2.3.4 have been applied to a single exponential distribution and mixtures. Table 2-4-1, 2-4-2, and 2-4-3 gives the results of comparisons of these methods and the estimators transformed to Gumbel distributions are displayed in Figure 2-4. We have tried several examples and selected the cases which all three methods worked properly. We have found in many cases the estimators by Rider were complex-valued or negative-valued, and often the EM algorithm did not converge even with good initial values under relatively generous convergence tolerances. The LSMDE sometimes falsely identifies underlying mixtures as a single exponential distribution, even though the mixtures considered were well separated.

Data were simulated from the standard exponential distribution. Even though the simulated data were from a single exponential distribution, the method of moments and the MLE gives almost identical density estimates, while the LSMDE gives two location estimates, 1.622120 and 1.447614, with the mixing weight estimates, 0.811996 and 0.188004 respectively, which results in the density estimates shifted toward the right-hand side slightly (Figure 2-4 (a)). The LSMDE shows the underlying density function might be a single exponential distribution or a mixture of two close exponential distributions without knowing the number of components beforehand.

Two hundreds random variates were generated from the mixture of Exp(λ_1=10.0) and Exp(λ_2=0.1) with the mixing weight 0.5. The MLE gives the best approximation to the mixture (Figure 2-4 (b)). The estimate by the LSMDE overestimates the mode located in the left side, and underestimates the mode in the right considerably. The LSMDE suggests that the underlying distribution consists of three components, two located around X=15 with the proportion of 0.7, the other located around X=1.43 with the proportion of 0.3 approximately. From the small proportion and the location of the third component in the LSMDE (Table 2-4-2), the underlying density can be considered as a mixture of 2 exponentials. The method of moments gives the worst estimates in spite of the mathematical properties such as consistency. It could not locate the mode at the left-hand side properly and the mode at the right-hand side was underestimated.

The last data set was generated from the mixture of Exp(λ_1=30.0), Exp(λ_2=1.0), and Exp(λ_3=1/30) with the proportion of 1/3 respectively (n=200). Since the method of moments by Rider is only for the case of a mixture of 2 exponentials, the estimates are not displayed in Figure 2-4 (c). Also estimates by the EM algorithm for a mixture of 2 exponentials have been tried, but not displayed or tabled, because those estimates are inferior by failing to locate one of the well separated three modes due to the misspecification of the number of components. Again the MLE for the mixture of 3 exponentials yields the best approximation. The LSMDE identifies the underlying mixture with a mixture of four components. Since the last component has relatively small proportion compared to others, and locates close to the first component, it might be ignored or combined to the first component.

In summary, estimates by the EM algorithm almost always gave better approximations than any other two methods considered provided that the number of components is known. Performance of the method of moments was unpredictable in that it provides unacceptable estimates in many cases. The LSMDE yields less accurate estimates for the mixture problem than the MLE. In most cases, the LSMDE successfully provides the rough estimate of the number of components with proper location parameters of exponential mixtures and it can be used as the guideline for the identification of the number of components for other available mixture techniques.

Table 2-4-1 : Standard Exponential Distribution (n=200)

Parameter	True density	Moment	EM	LSMDE
π_1	1.0	0.989662	0.989130	0.811996
π_2	N/A	0.010338	0.010870	0.188004
λ_1	1.0	1.030497	1.001130	1.622120
λ_2	N/A	0.287256	1.302218	1.447614
μ_1	1.0	0.970406	0.998871	0.616477
μ_2	N/A	3.481217	0.767921	0.690792
α_1	0.0	-0.030041	-0.001129	-0.483734
α_2	N/A	1.247382	-0.246407	-0.369917

NOTE : $\lambda = 1/\mu$, $\alpha = \ln(\mu)$. Data set is generated by S-plus rexp(200,rate=1) with set.seed(142) . The EM algorithm converges after 54 iterations with the initial values of λ_1=1.1, λ_2=0.9, π_1=0.01, π_2=0.99, tolerance=1e-4.

Table 2-4-2 : Mixture of 2 exponentials

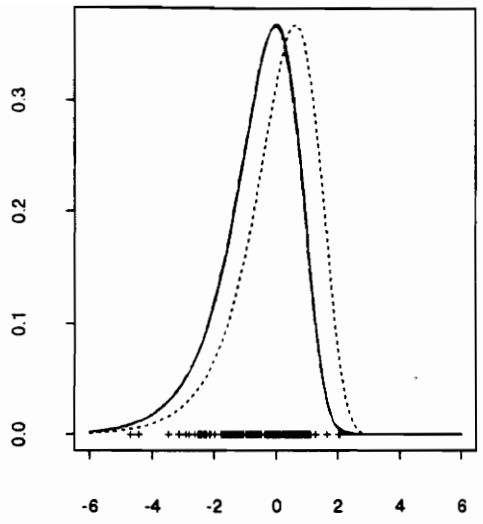
Parameter	True density	Moment	EM	LSMDE
π_1	0.5	0.719866	0.546818	0.603958
π_2	0.5	0.280134	0.453182	0.325098
π_3	N/A	N/A	N/A	0.070944
λ_1	10.0	0.917824	8.755867	15.296360
λ_2	0.1	0.077170	0.104133	0.143723
λ_3	N/A	N/A	N/A	14.730790
μ_1	0.1	1.089534	0.114209	0.065375
μ_2	10.0	12.958460	9.603138	6.957812
μ_3	N/A	N/A	N/A	0.067885
α_1	-2.302585	0.085750	-2.169724	-2.727615
α_2	2.302585	2.561749	2.262090	1.939865
α_3	N/A	N/A	N/A	-2.689940

NOTE : $\lambda = 1/\mu$, $\alpha = \ln(\mu)$. Data are generated by S-plus rexp.mixture2(200,0.5,0.5,10,0.1). The EM algorithm converges after 21 iterations with the initial values of λ_1=1.1, λ_2=0.2, π_1=0.55, π_2=0.45, tolerance=1e-6.

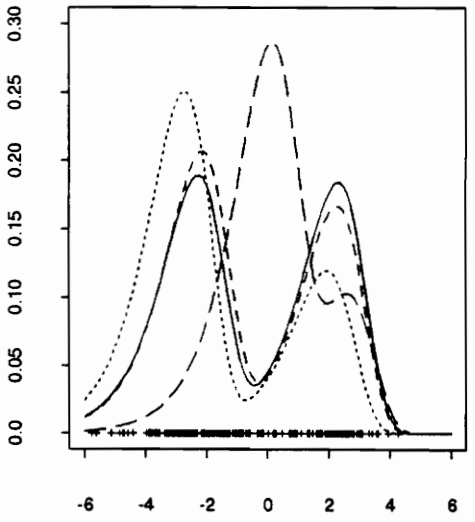
Table 2-4-3 : Mixture of 3 exponentials

Parameter	True density	Moment	EM	LSMDE
π_1	1/3	0.708977	0.368021	0.309279
π_2	1/3	0.291023	0.333243	0.409409
π_3	1/3	N/A	0.298736	0.248324
π_4	N/A	N/A	N/A	0.032988
λ_1	30.0	0.608574	38.9812771	41.38559
λ_2	1.0	0.032204	0.9232479	1.012563
λ_3	1/30	N/A	0.0374262	0.058477
λ_4	N/A	N/A	N/A	32.425960
μ_1	1/30	1.643187	0.025653	0.024163
μ_2	1.0	31.052070	1.083133	0.987593
μ_3	30.0	N/A	26.719260	17.100680
μ_4	N/A	N/A	N/A	0.025364
α_1	-3.401197	0.496638	-3.663081	-3.722933
α_2	0.0	3.435665	0.079858	-0.012485
α_3	3.401197	N/A	3.285385	2.839118
α_4	N/A	N/A	N/A	-3.674424

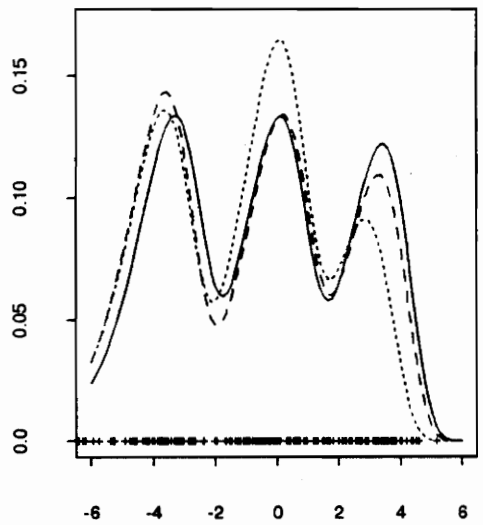
NOTE : $\lambda = 1/\mu$, $\alpha = \ln(\mu)$. Data are generated by S-plus rexp.mixture3(200,1/3,1/3,30,1,1/30,seed3) where seed3 = c(55, 2, 44, 30, 20, 56, 41, 1). The EM algorithm converges after 28 iterations with initial values of λ_1=1/25, λ_2=1.5, λ_3=19, π_1=1/3, π_2=1/3, π_3=1/3, and tolerance=1e-4.



(a) Gumbel(0)



(b) 0.5 (Gumbel(log(0.1))+Gumbel(log(10)))



(c) (Gumbel(log(30))+Gumbel(0)+Gumbel(log(1/30)))/3

Figure 2-4 : LSMDE for Gumbel mixtures. The solid line is true densities. The dotted line is LSMDE. The dashed line is MLE by the EM algorithm. The long dashed line in (b) is estimate by methods of moments. For (a), MLE and method of moments overlaped with true density.

Chapter 3

Multivariate Least Squares Mixture Decomposition Estimator

3.1 Extension of the LSMDE

As the dimension of data goes up, the visualization of the density functions becomes more challenging, and the intuition based on one or two dimensions may be misleading. Many difficulties in analyzing high-dimensional data are described as the curse of dimensionality. The curse of dimensionality describes the phenomenon - if the neighborhoods are local, then they are almost surely empty, whereas if a neighborhood is not empty, then it is not local [Scott (1992)]. For high-dimensional spaces, the tails or the relatively low density areas of a probability density function can play extremely important roles in the distribution. For example, in the ten-dimensional standard normal density, 99 % of the mass of the distribution is at points whose distance from the origin is greater than 1.6, whereas nearly 90 % of the one-dimensional normal density lies within the distance of 1.6. This empty space phenomenon leads us to the frustrating conclusion that ridiculously large numbers of observations are required to achieve moderate accuracy in MISE [See Table 4.2 of Silverman (1986)]. However, the nonparametric density estimation is still a good choice of techniques in a moderately low-dimensional space, even if the exploratory utility of the density estimates becomes less important than the univariate case. Density estimation in higher-dimensional space can be used as an intermediate step with other statistical methods such as kernel regression smoothing [Härdle (1991)] and discriminant analysis [McLachlan (1993)]. In this chapter, we

extend the LSMDE to the multivariate cases and show how it can be used to explore multi-dimensional densities.

3.1.1 Multivariate kernel density estimator

Let $\boldsymbol{X}$ be an $(n \times d)$ data matrix of random vectors $\underline{x} = (x_1, x_2, \cdots, x_d)^T$ where $\boldsymbol{X} = (\underline{x}_1, \underline{x}_2, \cdots, \underline{x}_n)^T$ is a random sample of size n from a multivariate density $f(\underline{x})$, of dimension d. Let x_{ij} denote the i-th observation of the j-th variable in $\boldsymbol{X}$. The multivariate kernel density estimator of $f(\underline{x})$ at $\underline{x} = \underline{y}$ is defined by

$$\hat{f}(\underline{y}) = \frac{1}{nh^d} \sum_{i=1}^{n} K\left(\frac{\underline{y} - \underline{x}_i}{h} \right) = \frac{1}{n} \sum_{i=1}^{n} K_h(\underline{y} - \underline{x}_i) \tag{3.1}$$

where $K(\underline{u})$ is a kernel function such that K: $R^d \rightarrow R^1$ satisfying

$$\int_{R^d} K(\underline{u}) d\underline{u} = 1. \tag{3.2}$$

Usually $K(\underline{u})$ will be a symmetric unimodal probability density function like univariate cases [Silverman (1986)]. In (3.1), the same bandwidth in each component is assumed for simplicity. For each component x_j of random variable, we can define a particular bandwidth h_j such that $\underline{h} = (h_1, h_2, \cdots, h_d)$. The multivariate product kernel density estimator is defined by

$$\hat{f}(\underline{y}) = \frac{1}{nh_1 \cdots h_d} \sum_{i=1}^{n} \prod_{j=1}^{d} K\left(\frac{y_j - x_{ij}}{h_j} \right) = \frac{1}{n} \sum_{i=1}^{n} \prod_{j=1}^{d} K_{h_j}(y_j - x_{ij}) \tag{3.3}$$

where $K_h(u) = h^{-1}K(u/h)$. The same symmetric univariate kernel, $K(u)$, is used in each dimension, but with a different bandwidth [Scott (1993)]. From now on, we denote the product kernel function by

$$\prod_{j=1}^{d} K_{h_j}(y_j - x_{ij}) = k_{\underline{h}}(\underline{y} - \underline{x}_i) \tag{3.4}$$

so that (3.3) becomes

$$\hat{f}(\underline{y}) = \frac{1}{n}\sum_{i=1}^{n} k_{\underline{h}}(\underline{y} - \underline{x}_i). \tag{3.5}$$

3.1.2 Definition

A multivariate extension of the LSMDE is easily derived from the straightforward extension of the univariate case. All the kernel function in the definition of the univariate LSMDE should be replaced by the product kernel function and the corresponding objective function is extended accordingly. Since the structure of the optimization is the same as the univariate case, we can still use the quadratic programming program from the previous chapter for the multivariate case with moderate modification.

Definition : Multivariate LSMDE of $f(\underline{x})$ at $\underline{x} = \underline{y}$

$\hat{f}(\underline{y})$ is defined as the multivariate least squares mixture decomposition estimator if for given an i.i.d random sample of size n from d-dimensional density function $f(\underline{x})$,

$\boldsymbol{X} = (\underline{x}_1, \underline{x}_2, \cdots, \underline{x}_n)^T$, the kernel function $K(u)$, and the fixed smoothing parameter h_j for x_j, for $j = 1, \cdots, d$

$$\hat{f}(\underline{y}) = \sum_{i=1}^{n} a_i \prod_{j=1}^{d} \frac{1}{h_j} K\left(\frac{y_j - x_{ij}}{h_j}\right) = \sum_{i=1}^{n} a_i k_{\underline{h}}(\underline{y} - \underline{x}_i) \tag{3.6}$$

where $\{a_i\}$, i=1,..,n minimize the objective function,

$$\hat{Q}[\hat{f}(\underline{y}), \underline{a}] = -\frac{2}{n}\sum_{i=1}^{n} \hat{f}(\underline{x}_i) + \int_{R^d} \hat{f}^2(\underline{y}) d\underline{y} \tag{3.7}$$

subject to $\sum_{i=1}^{n} a_i = 1$ and $a_i \geq 0$ for all $i = 1, \ldots, n$.

The objective function for multivariate cases, $\hat{Q}[\hat{f}(\underline{y}, \underline{a})]$, is readily obtained as follows. The linear term becomes

$$-\frac{2}{n}\sum_{i=1}^{n} \hat{f}(\underline{x}_i) = -\frac{2}{n}\sum_{i=1}^{n}\sum_{j=1}^{n} a_j k_{\underline{h}}(\underline{x}_i - \underline{x}_j) = \underline{a}^T \underline{d} \tag{3.8}$$

where

$$\underline{a} = (a_1, \quad a_2, \quad \cdots \quad a_n)^T,$$

$$\underline{d} = -\frac{2}{n}\left(\sum_{j=1}^{n} k_{\underline{h}}(\underline{x}_1 - \underline{x}_j), \quad \sum_{j=1}^{n} k_{\underline{h}}(\underline{x}_2 - \underline{x}_j), \quad \cdots \quad \sum_{j=1}^{n} k_{\underline{h}}(\underline{x}_n - \underline{x}_j)\right)^T.$$

The second term of $\hat{Q}[\hat{f}(\underline{y}, \underline{a})]$ (quadratic term) becomes

$$\int_{R^d} \hat{f}^2(\underline{y}) d\underline{y} = \int_{R^d} \sum_{i=1}^{n}\sum_{j=1}^{n} a_i a_j k_{\underline{h}}(\underline{y} - \underline{x}_i) k_{\underline{h}}(\underline{y} - \underline{x}_j)\, d\underline{y}$$

$$= \frac{1}{2}\underline{a}^T C \underline{a} \tag{3.9}$$

where

$$C = 2 \begin{bmatrix} k_{\underline{h}}^*(\underline{0}) & k_{\underline{h}}^*(\underline{x}_1 - \underline{x}_2) & \cdots & k_{\underline{h}}^*(\underline{x}_1 - \underline{x}_n) \\ k_{\underline{h}}^*(\underline{x}_2 - \underline{x}_1) & k_{\underline{h}}^*(\underline{0}) & \cdots & k_{\underline{h}}^*(\underline{x}_2 - \underline{x}_n) \\ \vdots & \vdots & \ddots & \vdots \\ k_{\underline{h}}^*(\underline{x}_n - \underline{x}_1) & k_{\underline{h}}^*(\underline{x}_n - \underline{x}_2) & \cdots & k_{\underline{h}}^*(\underline{0}) \end{bmatrix},$$

and

$$k_{\underline{h}}^*(\underline{x}_i - \underline{x}_j) = \int_{R^d} k_{\underline{h}}(\underline{y} - \underline{x}_i) k_{\underline{h}}(\underline{y} - \underline{x}_j) d\underline{y} = \int_{R^d} k_{\underline{h}}(\underline{z}) k_{\underline{h}}(\underline{z} - (\underline{x}_j - \underline{x}_i)\, d\underline{y},$$

the multivariate convolution of $k_{\underline{h}}(\underline{y} - \underline{x}_i)$ and $k_{\underline{h}}(\underline{y} - \underline{x}_j)$. (3.10)

Furthermore, all conjectures about the univariate LSMDE can be applied to multivariate cases.

3.2 Examples : Multivariate Cases

In this section, we test the multivariate LSMDE on some bivariate cases and a 4-dimensional case (the Iris data). The kernel function used is the multivariate standard normal density. Followings are the component of the quadratic programming.

$$k_{\underline{h}}(\underline{y}-\underline{x}_i)=\prod_{j=1}^{d}K_{h_j}(y_j-x_{ij})$$
$$=\frac{1}{(2\pi)^{d/2}|\Sigma|^{1/2}}\exp\left[-\frac{(\underline{y}-\underline{x}_i)^T\Sigma^{-1}(\underline{y}-\underline{x}_i)}{2}\right], \tag{3.11}$$

$$k^*_{\underline{h}}(\underline{x}_i-\underline{x}_j)=\int_{R^d}k_{\underline{h}}(\underline{y}-\underline{x}_i)k_{\underline{h}}(\underline{y}-\underline{x}_j)d\underline{y}$$
$$=\frac{1}{(2\pi)^{d/2}|\Sigma_*|^{1/2}}\exp\left[-\frac{(\underline{y}-\underline{x}_i)^T\Sigma_*^{-1}(\underline{y}-\underline{x}_i)}{2}\right], \tag{3.12}$$

$$\Sigma=diag(h_1^2,h_2^2,\cdots,h_d^2), \tag{3.13}$$

$$\Sigma_*=diag(2h_1^2,2h_2^2,\cdots,2h_d^2). \tag{3.14}$$

Note that (3.11) and (3.12) are multivariate normal density functions of $\underline{y}$ with mean vector $=\underline{x}_i$, covariance matrix $=\Sigma=diag(h_1,h_2,\cdots,h_d)$ and mean vector $=\underline{x}_i$, covariance matrix $=\Sigma_*=diag(2h_1,2h_2,\cdots,2h_d)$ respectively. Further simplification can be done if the same bandwidth is used in each direction.

3.2.1 Bivariate standard normal density

Like the univariate case, we start the discussion by examining the bivariate standard normal distribution. A random sample of size n=200 was generated using IMSL/RNMVN with random seed of 123457. We expect remaining one or two positive weights after the optimization to be close to 1 around the (0,0) for a proper smoothing parameter. The smoothing parameter from 0.6 to 1.6 is tested (See Table 3-1, Fig 3-1 (A,

B)). First the number of positive weights decreases rapidly from 20 to 2 as h increases. There are three positive weights that are greater than 0.05 after the optimization for h=1.0. The maximum of positive weights ($\underline{a}_{118}$=0.590045) locates at $\underline{x}_{118}$= (0.248213, 0.281438) which is not so close to the true mode (0,0). However, other two remaining positive weights locate the opposite side of $\underline{x}_{118}$ across (0,0). As h exceeds 1.0, remaining two positive weights locate at $\underline{x}_{94}$= (-0.08397,0.155762) and $\underline{x}_{105}$ = (0.024371, -0.06392), and we observe the positive weight is getting close to 1 moving toward $\underline{x}_{105}$ = (0.024371, -0.06392). Even though the LSMDE suggests a mixture of two normal kernel functions until h reaches 1.6, the location parameter of the kernel function is getting closer to (0,0). The maximum difference between the true density and LSMDE was less than 0.05 at h=1.0 and more than 0.05 at h=1.2 indicating oversmoothing. The LSMDE detects the unimodality of the underlying density function instantly. After h increases beyond 1.0, The LSMDE underestimates the true mode severely.

Table 3-1 : Positive weights for the simulated bivariate standard normal distribution

Data : Random sample of size n=200 from $N(\underline{0}, I)$

ID	X1	X2	H=0.6	H=0.8	H=1.0	H=1.2	H=1.4	H=1.6
4	-2.22547	1.192085	0.006858	0	0	0	0	0
26	-1.16396	-0.41132	0.070359	0	0	0	0	0
34	-1.00839	0.042962	0.067467	0	0	0	0	0
42	-0.91374	-0.10908	0.134542	0.225302	0	0	0	0
43	-0.90274	-0.31529	0	0.086598	0	0	0	0
47	-0.82313	-1.52534	0.005815	0	0	0	0	0
56	-0.59586	-0.34319	0	0	0.20346	0	0	0
57	-0.58112	-0.07341	0	0	0.137504	0	0	0
58	-0.58105	1.339761	0.020999	0	0	0	0	0
67	-0.45987	-1.74192	0.007165	0	0	0	0	0
69	-0.45365	1.53215	0.039434	0	0	0	0	0
82	-0.29811	-1.35638	0.089535	0	0	0	0	0
94	-0.08397	0.155762	0	0	0.00965	0.413606	0.313158	0.234007
95	-0.07449	2.042981	0	0.012761	0	0	0	0
103	0.007651	-1.48765	0	0.118905	0.020892	0	0	0
105	0.024371	-0.06392	0	0	0	0.586394	0.686842	0.765993
113	0.170623	2.670175	0	0.003058	0	0	0	0
117	0.245023	0.525694	0.173547	0.344102	0	0	0	0
118	0.248213	0.281438	0	0	0.590045	0	0	0
123	0.340062	-1.38054	0	0.029691	0.038449	0	0	0
125	0.357409	0.63599	0	0.024219	0	0	0	0
127	0.365883	-1.37565	0.053472	0	0	0	0	0
130	0.403497	2.876993	0.006449	0	0	0	0	0
132	0.417717	0.470228	0.127873	0.030834	0	0	0	0
142	0.550348	-1.45466	0.024556	0	0	0	0	0
149	0.65819	2.678947	0.003654	0	0	0	0	0
159	0.82017	1.905603	0.027787	0	0	0	0	0
175	1.074151	-0.13899	0	0.059257	0	0	0	0
176	1.101546	-0.23207	0.052643	0.065271	0	0	0	0
179	1.149499	-0.35696	0.075848	0	0	0	0	0
191	1.518034	-2.53905	0.003717	0	0	0	0	0
195	1.790396	0.442553	0.008278	0	0	0	0	0
#(positive weights)			20	11	6	2	2	2
Minimized Q[f,a]			-0.07827	-0.07546	-0.07384	-0.07142	-0.06476	-0.05703

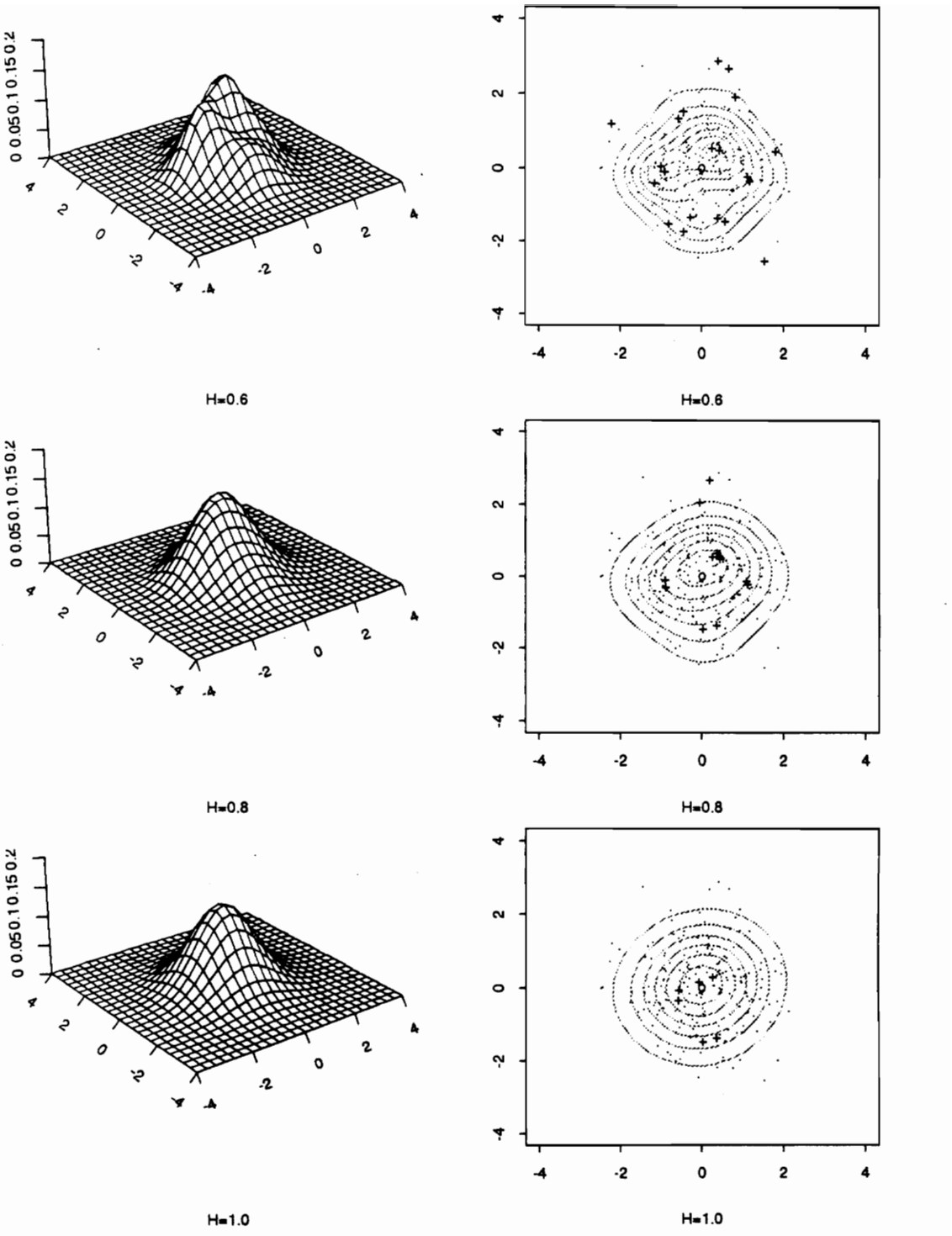


Figure 3-1 (A) : LSMDE for $BVN(\underline{0}, I)$ (n=200). "o" denotes (0,0) and "+" denotes positive weights.

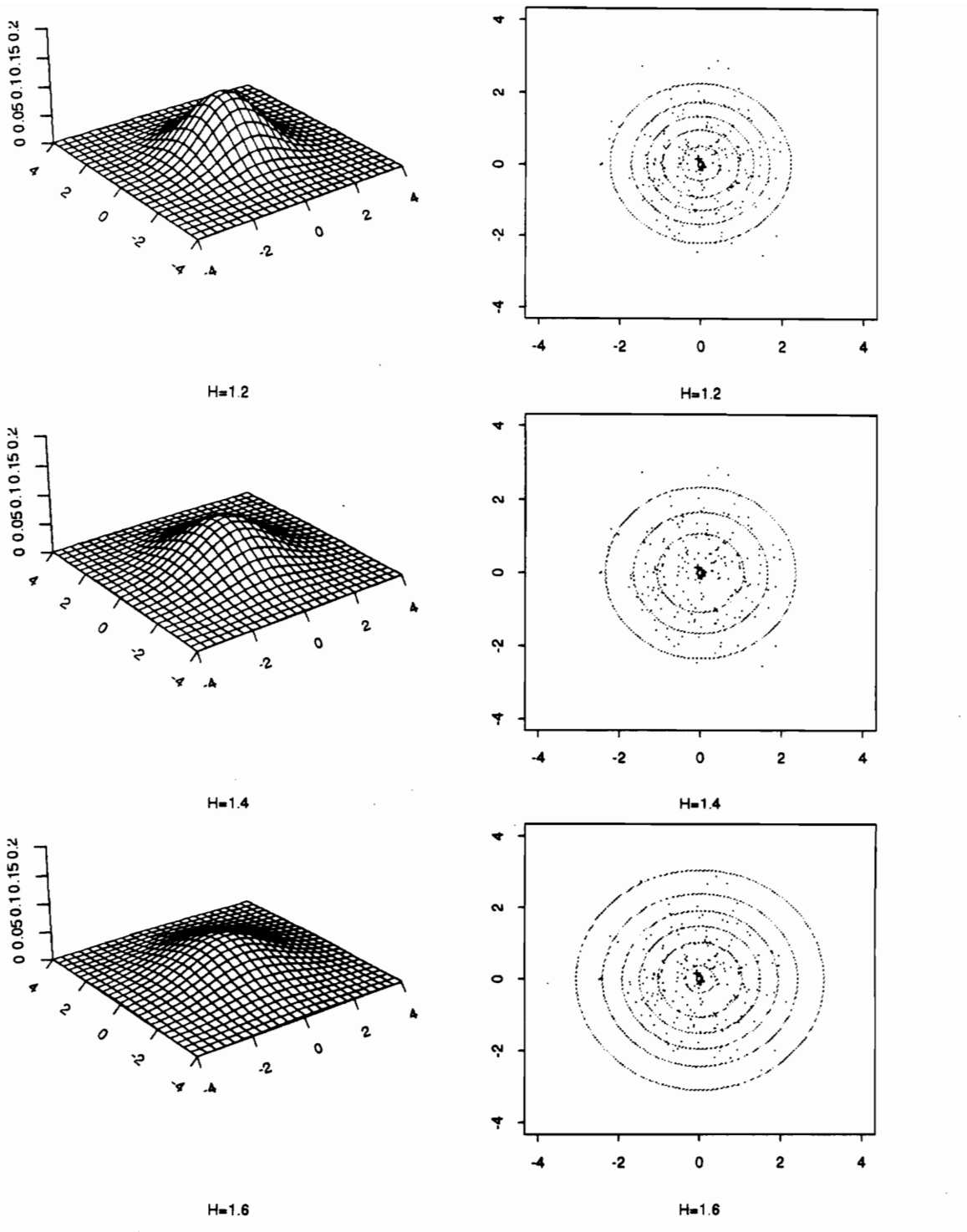


Figure 3-1 (B) : LSMDE for $BVN(\underline{0}, I)$ (n=200). "o" denotes (0,0) and "+" denotes positive weights.

3.2.2 Mixture of 2 bivariate normal densities

Next example is a mixture of 2 bivariate normal densities with the component parameters, $\underline{\mu}_1 = (-1.5, 0)^T$, $\underline{\mu}_2 = (1.5, 0)^T$, $\Sigma_1 = \Sigma_2 = I_2$, and the mixing proportion p of 0.5. This mixture density is symmetric around (0,0) with two well-separated modes at its component means. Two hundred data points were generated from this mixture using IMSL/RNMVN with the random seed of 75432 and the smoothing parameter from 0.6 to 1.6 were tested (see Table 3-2, Fig 3-2 (A, B)). Eighteen data points among 200 observations have positive weights when h=0.6. For h=1.0, we observe several clusters of positive weights. While some of them locate around (-1.5,0) and (1.5,0), it is not so clear that significant weights concentrate on the true modes. While the mode at the left side is overestimated, the right side mode is underestimated. As h increases above 1.0, two prominent clusters appear around the true modes (the sum of weights around (-1.5,0) $\approx$ 0.55, the sum of the weights around (1.5,0) $\approx$ 0.43 when h=1.2). While the LSMDE becomes unimodal as h approaches 1.4, the cluster of positive weights still suggests two well-separated components. Note that the significant weight ($\underline{a}_{84}$=0.237626) suddenly appears at $\underline{x}_{84}$= (-0.59557, 0.245469) when h=1.6. This indicates that the LSMDE tends to be unimodal due to oversmoothing. Sudden appearance of positive weights could be misleading and it should be examined with caution whether the weight really comes from the component or just from the oversmoothing process.

Table 3-2 : Positive weights for the simulated bivariate normal mixture distribution

Data : Random sample of size n = 200 from $\mu_1 = (-1.5,0)'$, $\mu_2 = (1.5,0)'$, $\Sigma_1 = \Sigma_2 = I$ with mixing proportion = 0.5

ID	X1	X2	H=0.6	H=0.8	H=1.0	H=1.2	H=1.4	H=1.6
10	-2.65057	-0.57195	0.043529	0	0	0	0	0
17	-2.5484	-0.33505	0.101378	0	0	0	0	0
19	-2.47062	1.351999	0.001453	0	0	0	0	0
21	-2.3871	-0.34037	0	0.155496	0	0	0	0
23	-2.33526	-0.47851	0	0.030158	0.079706	0	0	0
33	-1.83864	-0.1574	0	0.006539	0.14088	0	0	0
43	-1.64941	-2.40744	0.015142	0	0	0	0	0
46	-1.54467	0.355067	0.102606	0	0	0	0	0
50	-1.3517	0.133974	0	0	0	0.098461	0	0
52	-1.32283	0.161023	0	0	0	0.382891	0.072122	0
57	-1.19413	0.072885	0	0	0.062648	0.074873	0.499047	0.379983
62	-1.10565	-0.84134	0	0.041962	0	0	0	0
63	-1.08713	-2.02697	0	0.002415	0	0	0	0
68	-0.95029	-0.75554	0.094714	0	0	0	0	0
70	-0.90122	0.50258	0.000571	0.204182	0.160212	0	0	0
74	-0.79273	0.752294	0	0.018174	0	0	0	0
80	-0.69657	0.826246	0.021183	0	0	0	0	0
84	-0.59557	0.245469	0	0	0	0	0	0.237626
86	-0.56133	0.503557	0	0	0.149631	0	0	0
88	-0.51024	0.830284	0.172729	0	0	0	0	0
89	-0.50635	0.68626	0	0.105908	0	0	0	0
112	0.368347	-2.25677	0.002162	0	0	0	0	0
129	0.874886	-0.31379	0.094997	0	0	0	0	0
135	1.09491	-0.34926	0.042719	0.015708	0	0	0	0
136	1.114721	-0.42367	0	0.092713	0	0	0	0
145	1.484783	0.172186	0	0	0	0	0	0.38239
147	1.581942	0.155545	0	0	0	0.367352	0.428831	0
152	1.706248	1.165544	0.054542	0.09731	0.018077	0	0	0
154	1.725077	1.511708	0.008378	0	0	0	0	0
156	1.766789	-0.13728	0	0	0.187996	0.019863	0	0
158	1.821635	0.756074	0	0	0.111324	0	0	0
162	1.893078	-0.49658	0	0.056833	0.080373	0	0	0
165	1.994418	1.180799	0.090694	0.060772	0	0	0	0
169	2.075432	0.293497	0	0	0	0.056561	0	0
175	2.197685	-0.66237	0.042615	0.031747	0	0	0	0
178	2.352874	-0.39836	0	0.040584	0.009153	0	0	0
179	2.395005	-0.69882	0.092482	0	0	0	0	0
185	2.526763	-0.50775	0	0.039499	0	0	0	0
197	3.384816	0.173184	0.018108	0	0	0	0	0
#(positive weights)			18	16	10	6	3	3
Minimized Q[f,a]			-0.04982	-0.04631	-0.04416	-0.04199	-0.03904	-0.03597

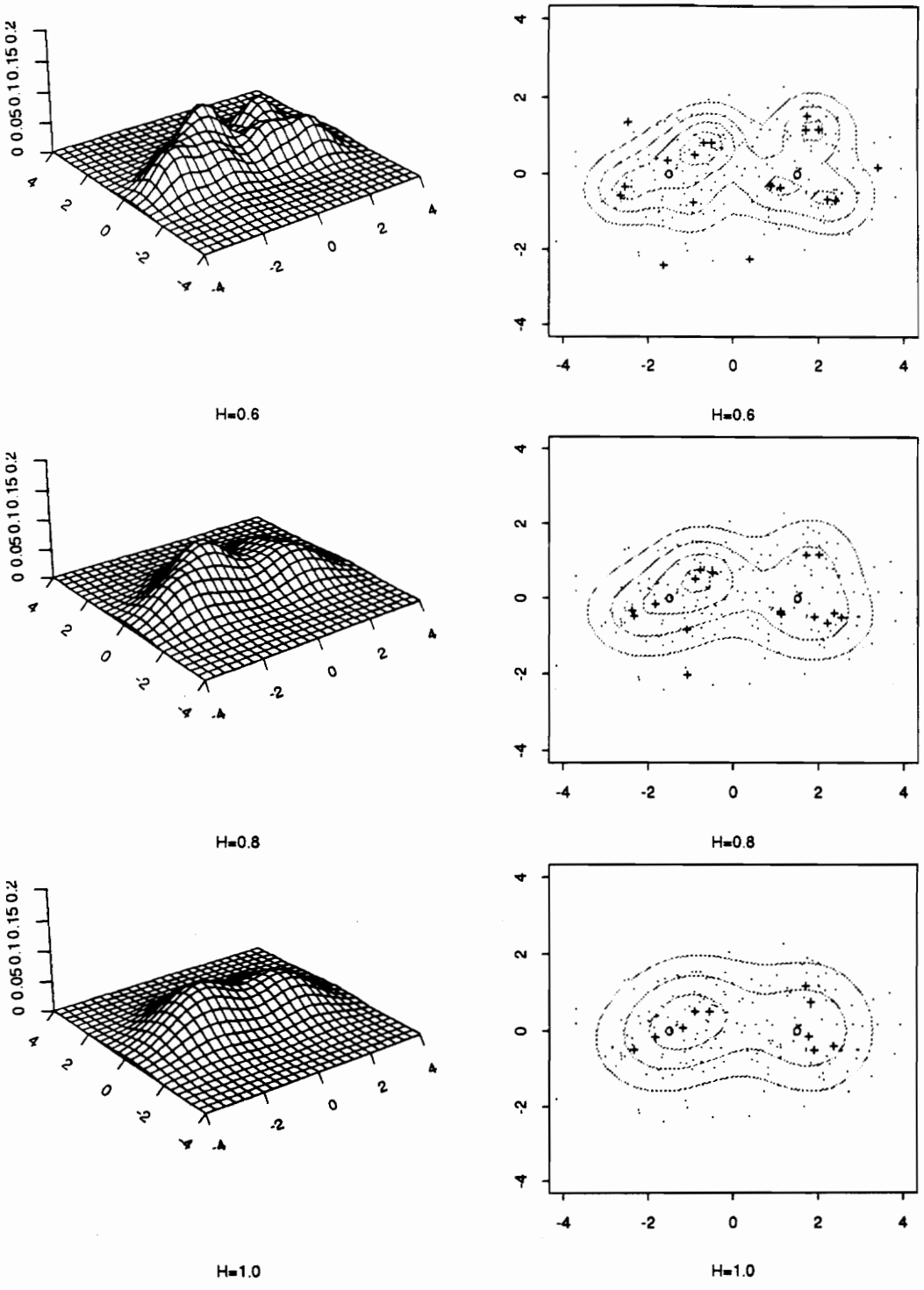


Figure 3-2 (A) : LSMDE for bivariate normal mixture with $\mu_1 = (-1.5,0)'$ $\mu_2 = (1.5,0)'$ $\Sigma_1 = \Sigma_2 = I$ p=0.5. "o" denotes (-1.5,0) and (1.5,0) and "+" denotes positive weights.

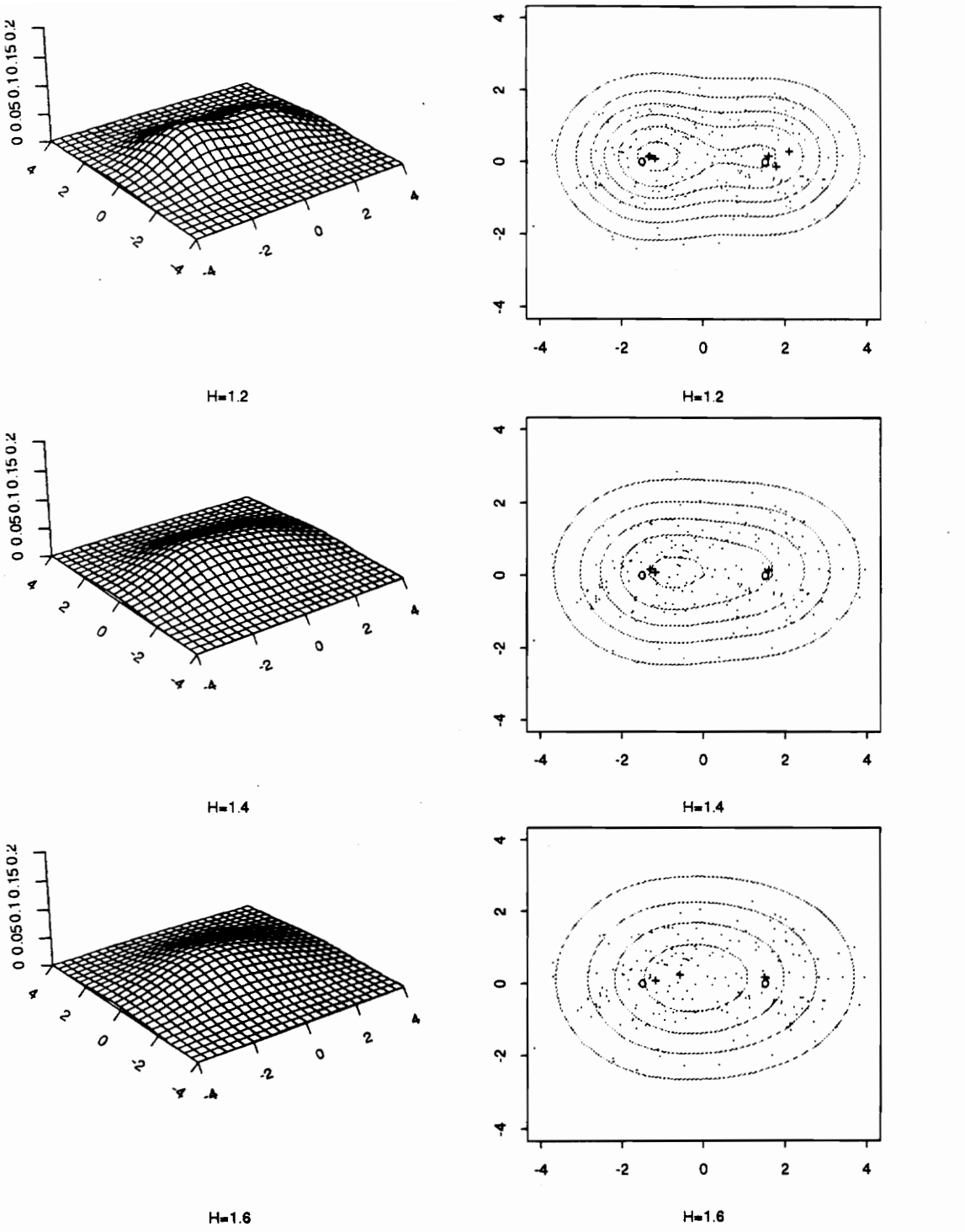


Figure 3-2 (B) : LSMDE for bivariate normal mixture with $\mu_1 = (-1.5,0)'$ $\mu_2 = (1.5,0)'$ $\Sigma_1 = \Sigma_2 = I$ p=0.5. "o" denotes (-1.5,0) and (1.5,0) and "+" denotes positive weights.

3.2.3 Mixture of 3 bivariate normal densities : Scott density

So far we have examined distributions where the true mode(s) of underlying densities always correspond(s) to their component means. In this section we consider an example to illustrate how the LSMDE identifies the component of a mixture distribution.

Consider the mixture of three normal distributions with the following component parameter suggested by Scott.

$$\underline{\mu}_1 = (1,0)^T,\ \underline{\mu}_2 = (-\frac{1}{2}, \frac{\sqrt{3}}{2} \approx 0.866025)^T,\ \underline{\mu}_3 = (-\frac{1}{2}, -\frac{\sqrt{3}}{2})^T,$$

$$\Sigma_1 = \Sigma_2 = \Sigma_3 = diag(0.7355^2, 0.7355^2),$$

with the mixing proportion $p_1 = p_2 = p_3 = \frac{1}{3}$. This mixture of three normal distributions has 4 modes, three located close, not exactly, to its component mean vectors and the remaining one at (-0.5,0). Each component mean vector forms a vertex of the equilateral triangle. Since the mixture is almost flat over this triangle, the mode at (-0.5,0) is difficult to detect. The mode at (-0.5,0) is generated indirectly by the effect of 3 component densities. We are interested in how many components the LSMDE identifies for this particular example. Positive weights and estimates are displayed in Table 3-3 and Fig 3-3 (A,B). A random sample of size n=300 was generated by IMSL/RNMVN with the random seed of 754321. We observe 3 or 4 clusters of positive weights when h ranges from 0.8 to 1.2. When h = 1.0, positive weights that are greater than 0.05 forms 3 or 4 clusters around each component mean and each cluster has approximately the weight of 0.3 (the sum of weights is 0.356 around (1,0), 0.305 around (-0.5,0.87), and 0.27 around (-0.5,-0.87)). The estimate suggests that the underlying density function consists of 3 or 4 components for h=0.8 and 1.0 (≈0.7355). As h increases beyond 1.2,

clusters of positive weight get closer around the true mode located at (-0.5, 0) due to the flatness of the mixture over the isosceles ignoring the other 3 modes .

3.2.3 Cholesterol lipid data

The data set we will review next comes from a study of 320 males suffering from chest pain [Scott et al. (1978)]. Concentration of plasma cholesterol and plasma triglcerids in 320 chest pain male patients are recorded. Scott revealed the bimodality of the underlying distribution using the average shifted histogram (ASH) technique and divided the population into two groups, and an interesting clinical difference was found between two groups. A broad range of smoothing parameters (from 30 to 55 by the increment of 5) has been applied to this data. Although the LSMDE has been spiky for the small bandwidths (from 30.0 to 35.0), it strongly shows the bimodality (Fig 3-4 (A)). Additionally, we can find a cluster of small positive weights remaining strongly at the upper portion of each graph (Fig 3-4 (B)). When h=55, we find only three positive weights left and two of 3 remaining positive weights are located in the lower part of the plot (around (200, 200)). The other stays at (229, 296) which is small (0.091605), but still well separated from the cluster near (200, 200).

Table 3-3 : Positive weights for the simulated bivariate normal mixture distribution

Data : Random sample of size n = 300 from Scott density, where

$$\mu_1 = (1,0)', \mu_3 = (-\frac{1}{2}, \frac{\sqrt{3}}{2})', \mu_3 = (-\frac{1}{2}, -\frac{\sqrt{3}}{2})', \ \Sigma_1 = \Sigma_2 = 0.7355^2 I$$

with mixing proportion = 1/3

ID	X1	X2	H=0.8	H=1.0	H=1.2	H=1.4	H=1.6
2	-3.04818	-1.50454	0.001291	0	0	0	0
3	-2.96864	-1.77129	0.003062	0	0	0	0
15	-1.99256	-1.38833	0.008218	0	0	0	0
20	-1.83614	2.194743	0.003212	0	0	0	0
29	-1.57283	2.100071	0.027129	0	0	0	0
31	-1.55305	2.641151	0	0.007363	0	0	0
40	-1.31831	-0.91854	0.08989	0	0	0	0
47	-1.27012	-1.03123	0	0.021038	0	0	0
54	-1.1573	3.296149	0	0.005118	0	0	0
66	-1.05581	-0.92281	0	0.087104	0	0	0
68	-1.01547	-0.61932	0.007645	0	0	0	0
79	-0.87059	0.690476	0.057274	0	0	0	0
83	-0.84623	-1.33877	0	0	0.021384	0	0
84	-0.81164	0.46861	0.06261	0	0	0	0
94	-0.59911	0.737087	0.121476	0.304716	0.189958	0	0
104	-0.42036	3.379006	0.015149	0	0	0	0
117	-0.31515	-1.22945	0	0	0.098914	0	0
147	-0.05753	-0.18489	0	0	0.194182	0.402122	0.526791
149	-0.03627	-1.31726	0.028682	0.182973	0	0	0
150	-0.03541	-1.40727	0.164121	0	0	0	0
158	0.063059	-0.34057	0	0	0.01482	0.076016	0
163	0.113382	0.091986	0	0	0.099407	0.521862	0.473209
166	0.161316	1.188948	0.038323	0	0	0	0
167	0.172008	0.903904	0.005652	0	0	0	0
197	0.526193	-0.05836	0	0	0.381333	0	0
205	0.596657	-1.39463	0	0.010262	0	0	0
216	0.720515	0.147794	0.081583	0.213195	0	0	0
220	0.777136	0.280807	0.091345	0	0	0	0
223	0.788205	0.018984	0	0.025594	0	0	0
234	1.050042	-0.02971	0.123869	0.142637	0	0	0
246	1.217454	0.108146	0.005352	0	0	0	0
269	1.605525	-1.60784	0.012501	0	0	0	0
272	1.680439	-1.7423	0.025765	0	0	0	0
277	1.820271	0.482692	0.011817	0	0	0	0
288	2.087234	0.957336	0.009295	0	0	0	0
295	2.787859	0.419793	0.00474	0	0	0	0
#(positive weights)			24	10	7	3	2
Minimized Q[f,a]			-0.05078	-0.05008	-0.04957	-0.04868	-0.04545

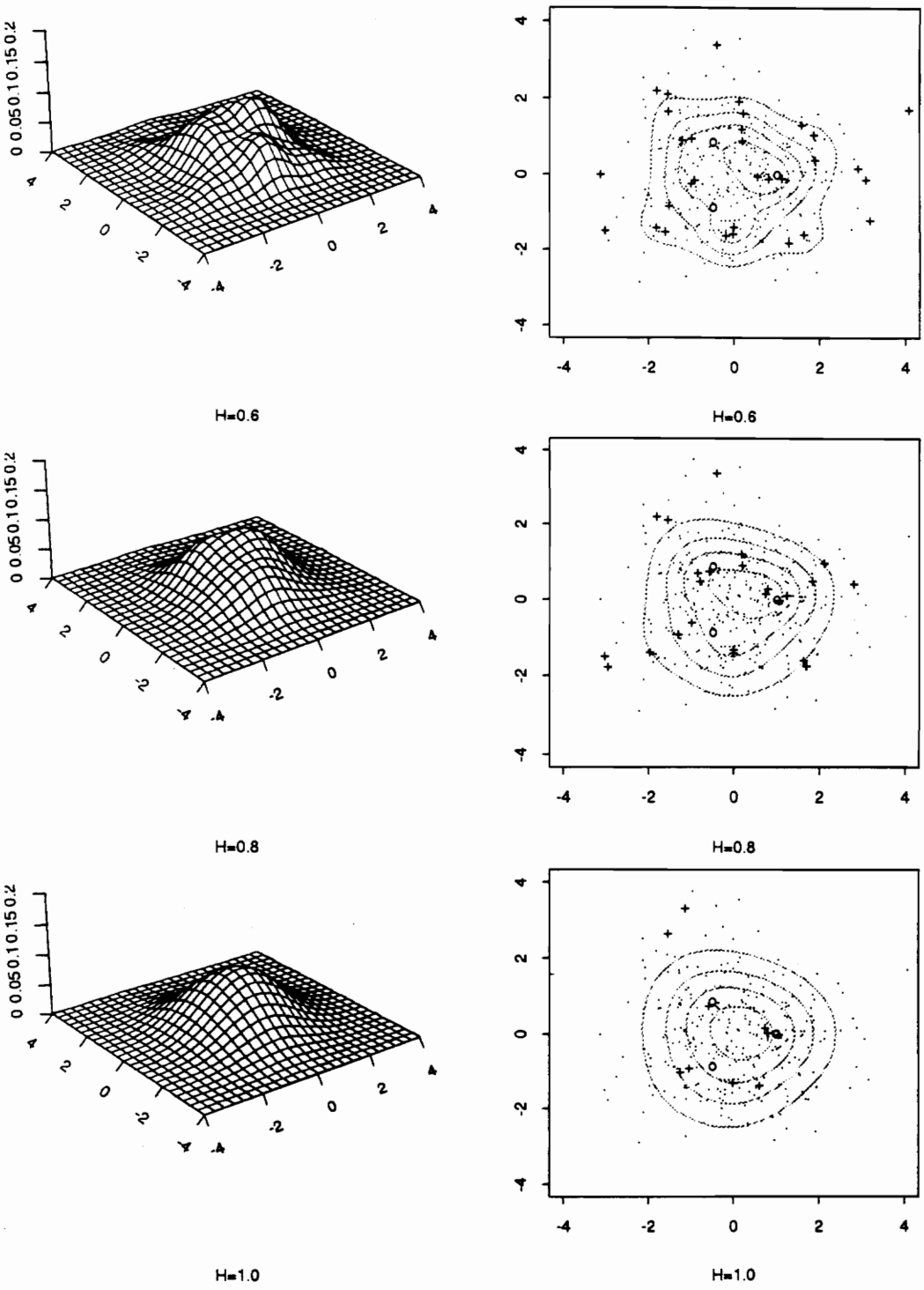


Figure 3-3 (A) : LSMDE for bivariate normal mixture with $\mu_1 = (1,0)'$, $\mu_2 = (-1/2, \sqrt{3}/2)'$, $\mu_3 = (-1/2, -\sqrt{3}/2)'$, $\Sigma_1 = \Sigma_2 = \Sigma_3 = 0.7355^2 I$ with proportion 1/3 . "o" denotes component means and "+" denotes positive weights.

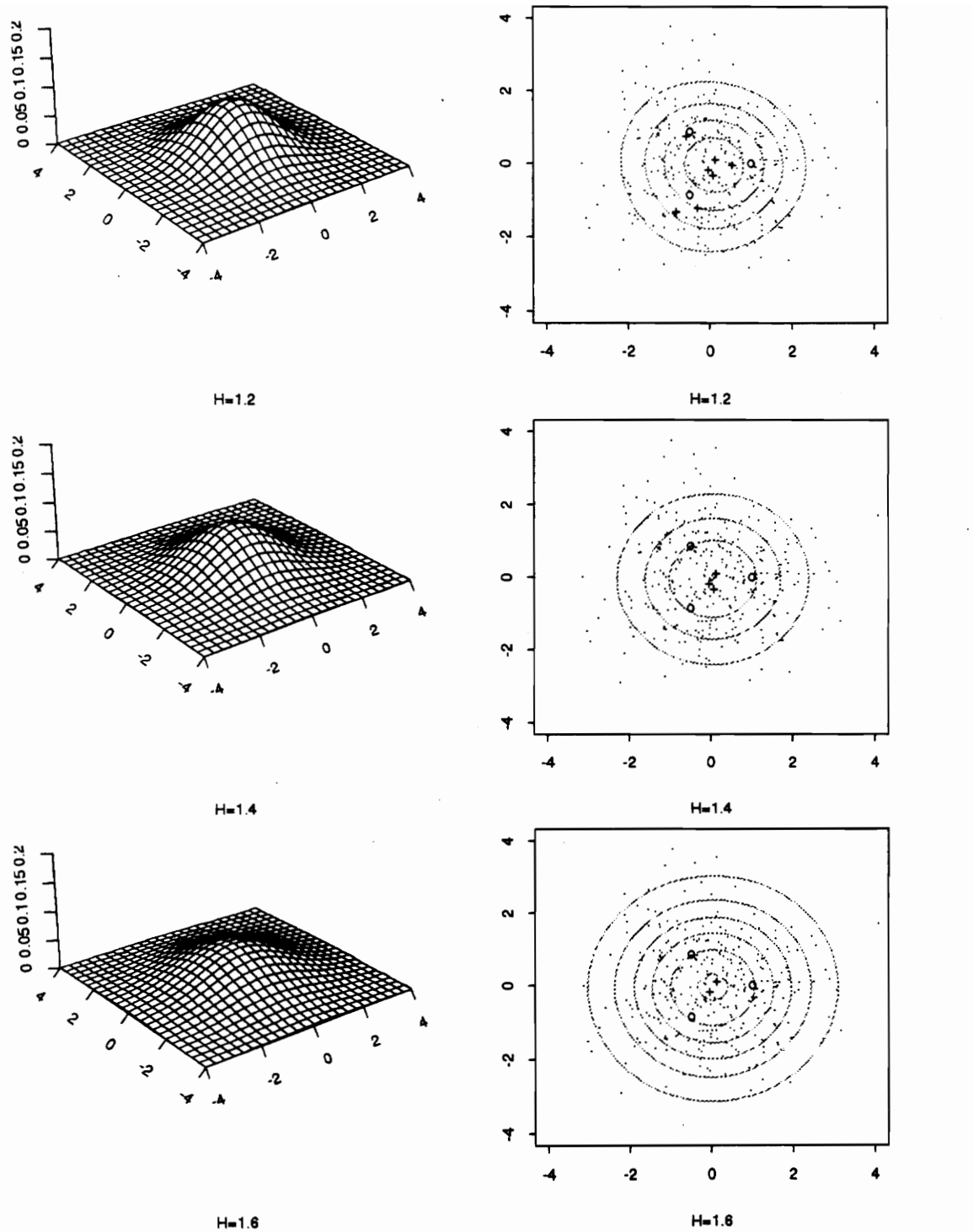


Figure 3-3 (B) : LSMDE for bivariate normal mixture with $\mu_1 = (1,0)'$, $\mu_2 = (-1/2, \sqrt{3}/2)'$, $\mu_3 = (-1/2, -\sqrt{3}/2)'$, $\Sigma_1 = \Sigma_2 = \Sigma_3 = 0.7355^2 I$ with proportion 1/3 . "o" denotes component means and "+" denotes positive weights.

Table 3-4 : Positive weights for the cholesterol lipid data (n=320)

ID	X1	X2	H=30.0	H=35.0	H=40.0	H=45.0	H=50.0	H=55.0
11	221	140	0	0	0	0	0.077916	0.17828
15	221	432	0.003944	0.011634	0.013214	0.008726	0	0
19	313	256	0.001081	0	0	0	0	0
30	284	245	0.011508	0.025759	0	0	0	0
52	386	162	0.002558	0.001425	0	0	0	0
53	236	152	0.306728	0.231906	0	0	0	0
55	188	220	0.045473	0	0	0	0	0
57	212	130	0	0	0	0.307635	0.192713	0
59	230	158	0	0.003891	0	0	0	0
64	297	232	0.0051	0	0.001053	0	0	0
65	232	328	0.007477	0.001487	0	0	0	0
73	417	198	0.000131	0	0	0	0	0
75	240	441	0.017568	0.013179	0.007604	0	0	0
76	191	115	0	0.164037	0	0	0	0
77	217	327	0.051763	0.051861	0.038506	0	0	0
78	208	262	0.065639	0.07989	0	0	0	0
80	191	115	0	0.164037	0	0	0	0
89	283	424	0	0.000042	0	0	0	0
105	237	400	0	0	0	0.005729	0	0
110	236	148	0.003948	0	0	0	0	0
116	306	408	0.007051	0.002291	0	0	0	0
120	285	930	0.000273	0	0	0	0	0
127	221	268	0	0	0	0.129764	0.046384	0
135	197	265	0.013812	0	0	0	0	0
156	191	233	0	0.018052	0	0	0	0
158	206	133	0	0	0	0.414838	0.595436	0.730115
161	219	267	0	0	0.086602	0	0	0
166	242	85	0.018123	0	0	0	0	0
181	246	87	0.004469	0	0	0	0	0
185	194	116	0	0	0.209791	0	0	0
195	191	149	0.020269	0	0	0	0	0
198	190	120	0	0.055289	0	0	0	0
201	105	36	0.002973	0	0	0	0	0
205	211	304	0	0	0	0.037435	0.008557	0
228	348	154	0.004037	0	0	0	0	0
229	194	400	0.005488	0	0	0	0	0
235	131	61	0.000662	0	0	0	0	0
271	198	124	0	0	0.28625	0.056564	0	0
279	229	296	0	0	0	0	0.078994	0.091605
280	232	583	0.00224	0	0	0	0	0
282	228	149	0	0.098699	0.244001	0.03931	0	0
283	187	115	0.255584	0.043912	0	0	0	0
293	259	240	0.029487	0	0	0	0	0
295	213	261	0	0.029622	0.053247	0	0	0
300	238	156	0	0	0.059733	0	0	0
309	218	567	0.003281	0.001228	0	0	0	0
316	225	240	0	0.001759	0	0	0	0
318	185	110	0.109331	0	0	0	0	0
		#(positive weights)	28	20	10	8	6	3
		Minimized Q[f,a]	-3.4E-05	-3.3E-05	-3.2E-05	-3.1E-05	-2.9E-05	-2.8E-05

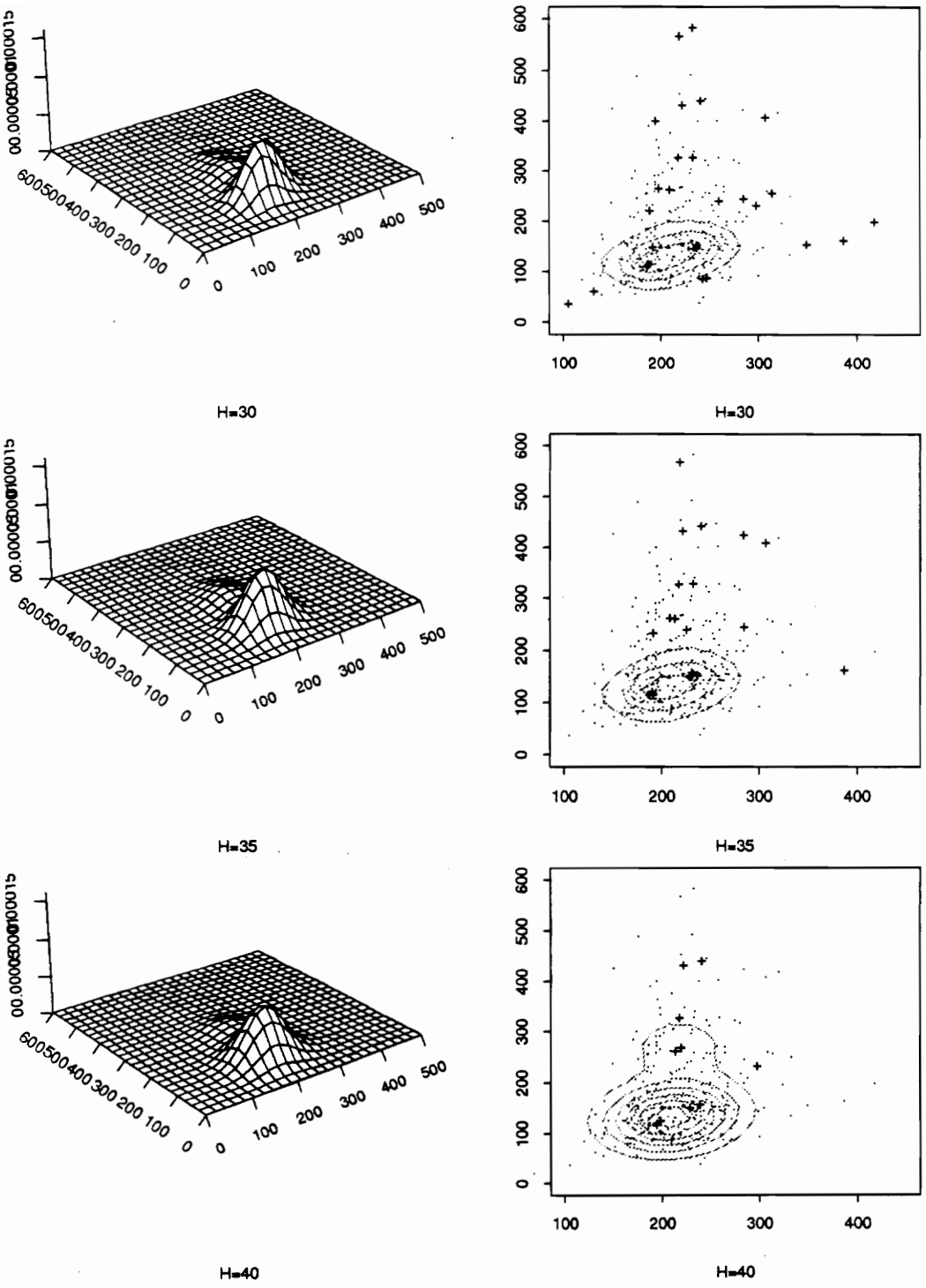


Figure 3-4 (A) : LSMDE for the cholesterol lipid data (n=320). "+" denotes positive weights.

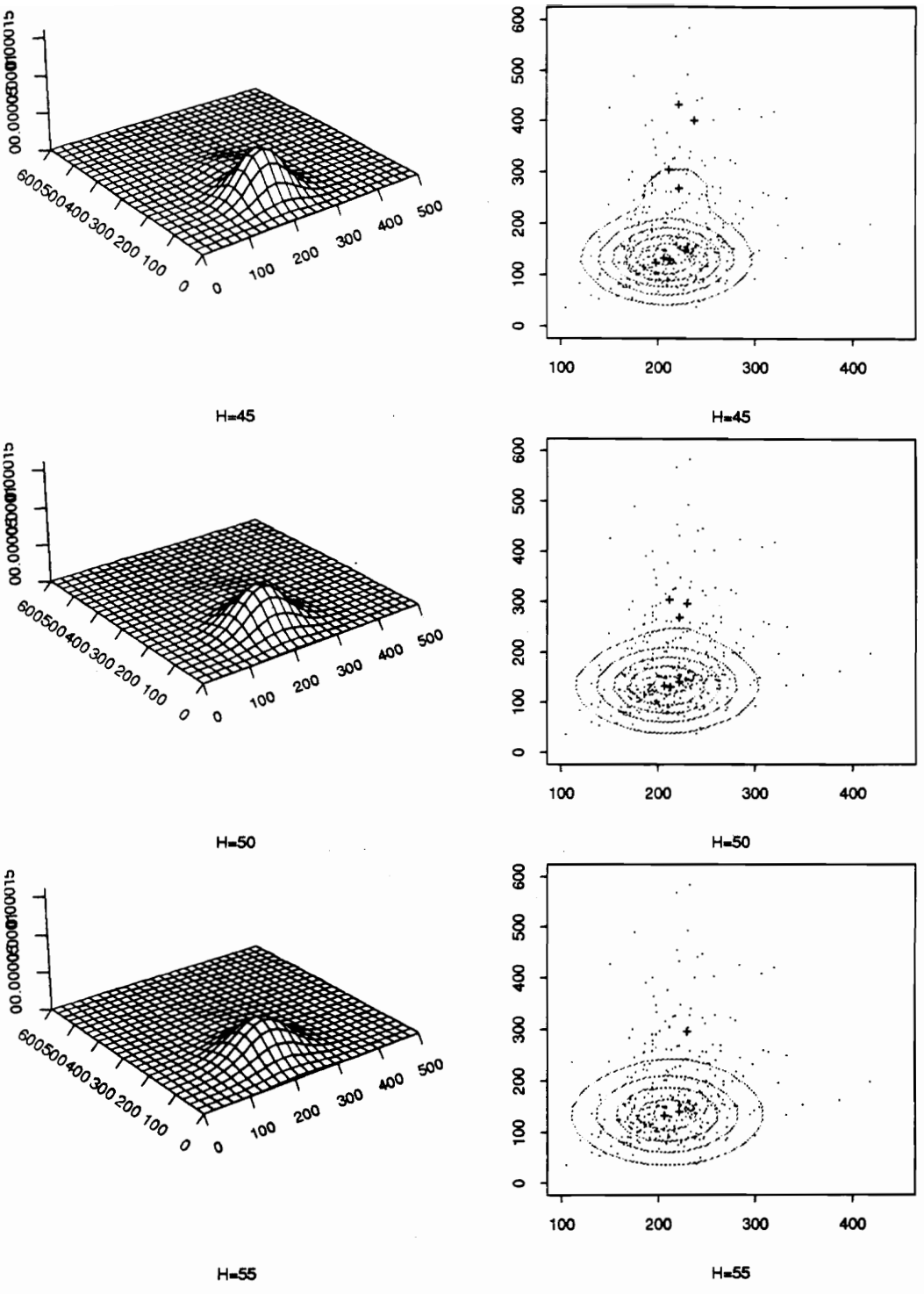


Figure 3-4 (B) : LSMDE for the cholesterol lipid data (n=320). "+" denotes positive weights.

3.2.4 Four-dimensional data set : the Iris Data

In this section, we demonstrate how the LSMDE can be used to explore high-dimensional data with the statistical graphics. Since the LSMDE decomposes the underlying density function to a finite mixture of the kernel functions, it can provide better understandings of the underlying density. Furthermore, it can enhance the perception of the higher dimensional data with graphical methods. The visualization of multivariate data becomes more difficult as the dimension goes up. It should be noted that some important aspects of multidimensional data can be overlooked by considering the data one variable at a time. For more than 3-dimensional data sets, the scatter plot matrix is the most commonly used graphical tool with dynamic display techniques like brushing and linking and can be enhanced with aid of the LSMDE.

The Iris data consist of 4 measurements, sepal length, sepal width, petal length, and petal width on 50 flowers from each of 3 species of Iris Setosa, Versicolor, and Virginica. This data set has been widely adapted for cluster analysis, discriminant analysis, and finite normal mixture models. First we start our analysis from the scatter plot matrix with the univariate kernel density estimates (Fig 3-5). The plots on the diagonal are the kernel density estimates of 4 different measurements. While sepal length and sepal width seem to be from unimodal symmetric densities, petal length and petal width show bimodality clearly. The bimodality or the cluster (Setosa versus. Versicolor and Virginica) of points can be easily detected even if we do not distinguish each species with the different symbols. A closer examination can separate Versicolor from Virginica when we use the different symbols for each species. Note that the univariate kernel density estimates for the sepal length and the sepal width cannot detect the fact that the data set is from 3 different species. For petal length and petal width, the univariate kernel

estimates suggest that the data might be from a mixture of two different components. However, three different components could not have been revealed without different labels used in the scatter plot matrix. From now on, we ignore the labels used for identifying species to see how the LSMDE detects 3 components in this example. The observations were classified in the order of Setosa (ID 1 - ID 50), Versicolor (ID 51 - ID 100), and Virginica (ID 101 - ID 150).

We have tried the smoothing parameter from 0.4 to 1.6. For h=0.4, eight positive weights (5 weights greater than 0.05) out of 150 observations spread all over the plots. As h increases to 0.6 and 0.8, only 3 positive weights remain and they are located at each species ($\underline{a}_8$=0.471166 at $\underline{x}_8$ (Setosa), $\underline{a}_{79}$=0.342314 at $\underline{x}_{79}$ (Versicolor), and $\underline{a}_{148}$= 0.186521 at $\underline{x}_{148}$ (Virginica) when h=0.6, see Fig 3-5 (A)). As h reaches 1.0, the positive weight for Versicolor ($\underline{a}_{64}$=0.043224) becomes smaller and less important indicating Versicolor and Virginica become indistinguishable. Note that while only two positive weights remain suggesting a mixture of Setosa and Virginica for h=1.2, a significant positive weight for Versicolor ($\underline{a}_{64}$=0.117) appears again as h increases to 1.4 (Fig 3-5 (B), (C)). This demonstrates that, as in univariate cases, the number of positive weights does not always decrease as h increases, and significant weights seem to move around as h changes. Finally as h reaches 1.6, all three positive weights locate in the middle of each plot. For h=0.6, 0.8, and 1.2, the LSMDE clearly identifies 3 components.

Table 3-5 : Positive weights for the Iris data (n = 150)

SL : Sepal Length, SW : Sepal Width, PL : Petal Length, PW: Petal Width

Setosa (ID1-ID50), Versicolor (ID51-ID100), Virginica (ID101-ID150)

ID	SL	SW	PL	PW	H=0.4	H=0.6	H=0.8	H=1.0	H=1.2	H=1.4	H=1.6
8	5	3.4	1.5	0.2	0.50121	0.471166	0.387112	0.289147	0.20626	0.127709	0
27	5	3.4	1.6	0.4	0	0	0	0	0	0	0.025579
64	6.1	2.9	4.7	1.4	0.108711	0	0.172191	0.043224	0	0.117338	0.191658
79	6	2.9	4.5	1.5	0.02614	0.342314	0	0	0	0	0.451885
100	5.7	2.8	4.1	1.3	0.153282	0	0	0	0	0	0
113	6.8	3	5.5	2.1	0.058279	0	0	0	0	0	0
127	6.2	2.8	4.8	1.8	0.035381	0	0.440696	0.667629	0.79374	0.754954	0.330877
128	6.1	3	4.9	1.8	0.012448	0	0	0	0	0	0
148	6.5	3	5.2	2	0.104549	0.186521	0	0	0	0	0
#(positive weights)					8	3	3	3	2	3	4
Minimized Q[f,a]					-0.19054	-0.0679	-0.02949	-0.01483	-0.00826	-0.00499	-0.00323

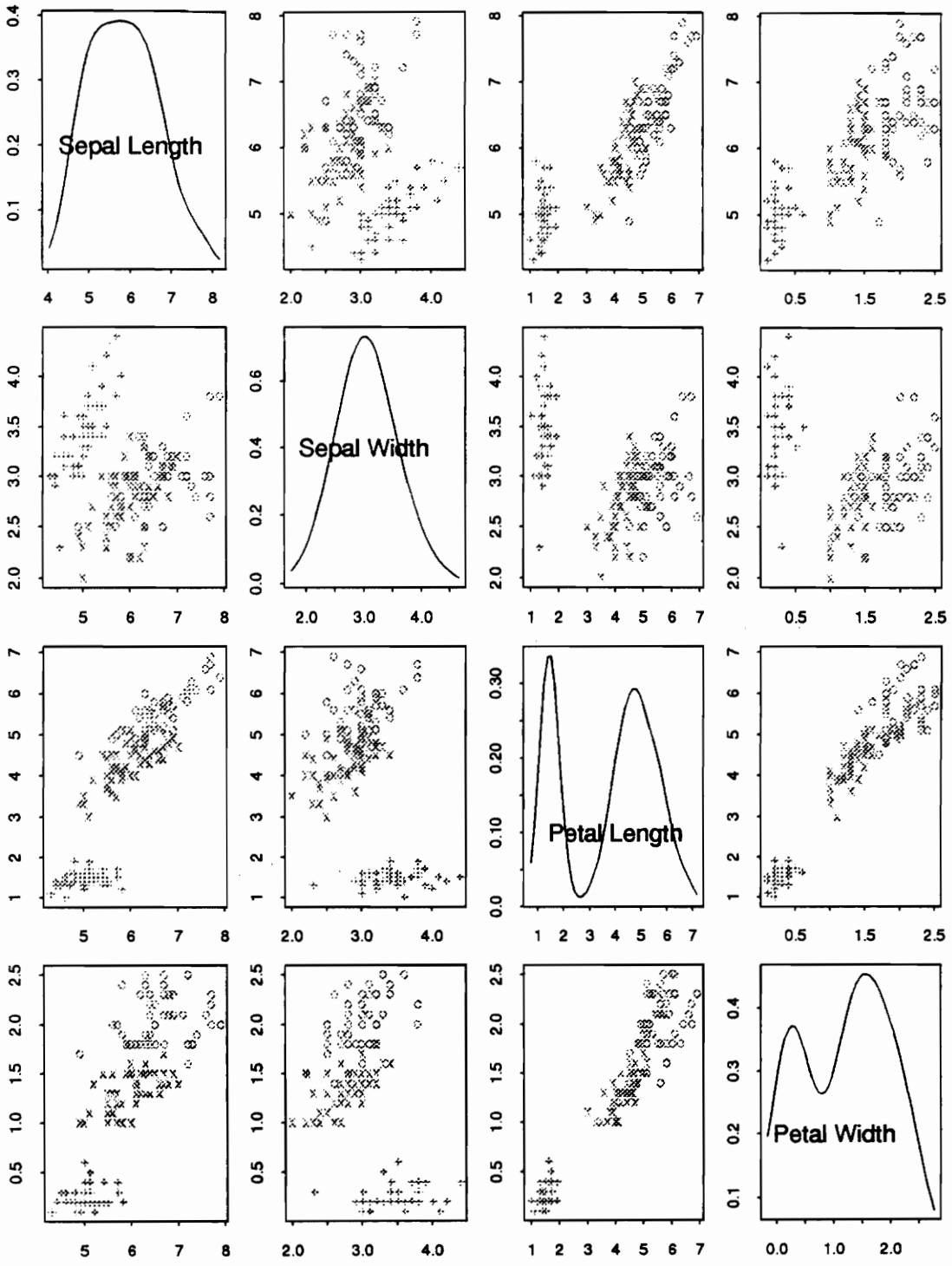


Figure 3-5 : Scatter plot matrix with univariate kernel density estimates for the Iris data. "+" denotes Setosa, "x" denotes Vericolor and "o" denotes Virginica.

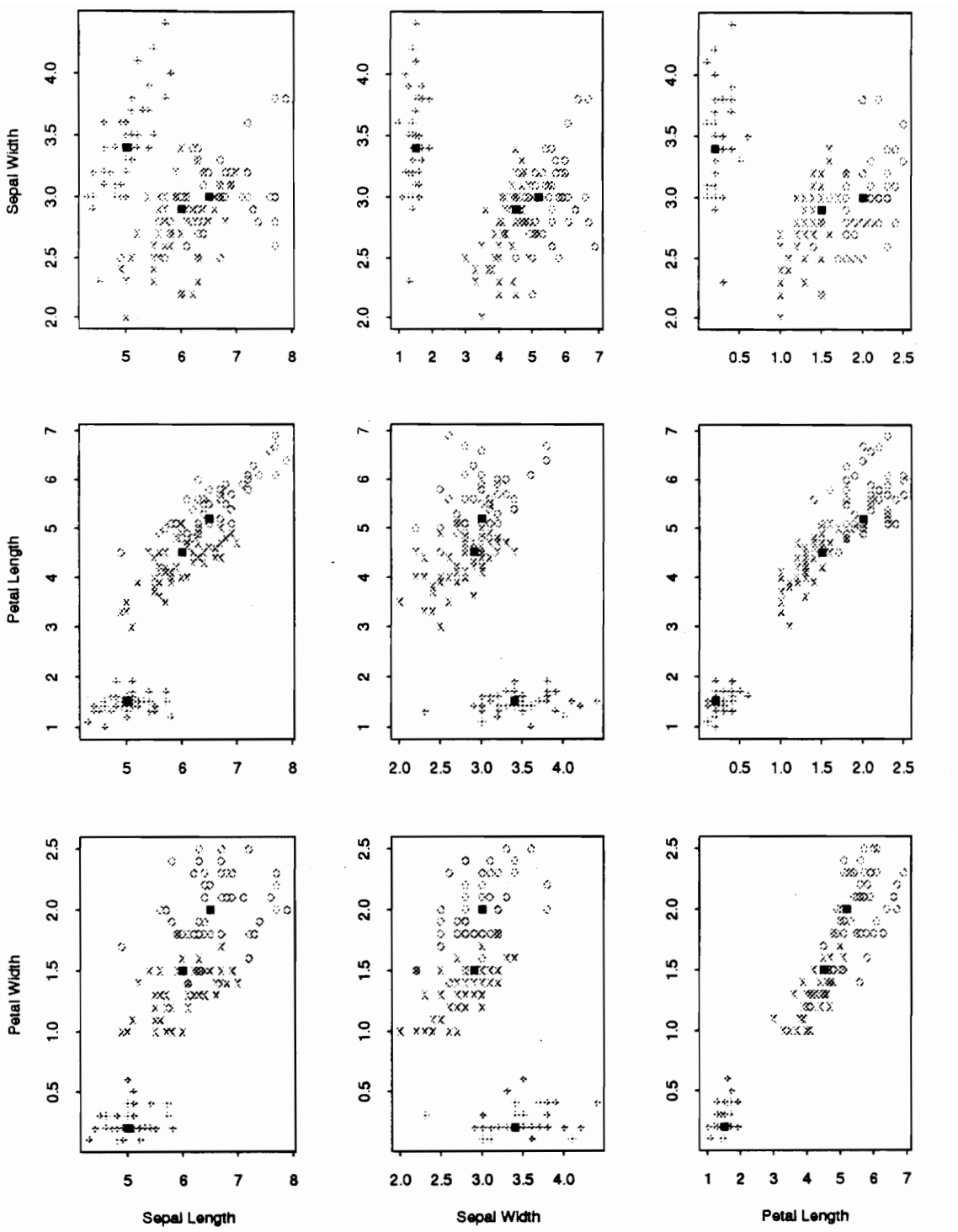


Figure 3-5 (A) : Positive weights by LSMDE for Iris data (H=0.6). "■" denotes positive weights, "+" denotes Setosa, "x" denotes Vericolor and "o" denotes Virginica. Positive weights are 0.47116 for ID8, 0.34231 for ID64, 0.18652 for ID148.

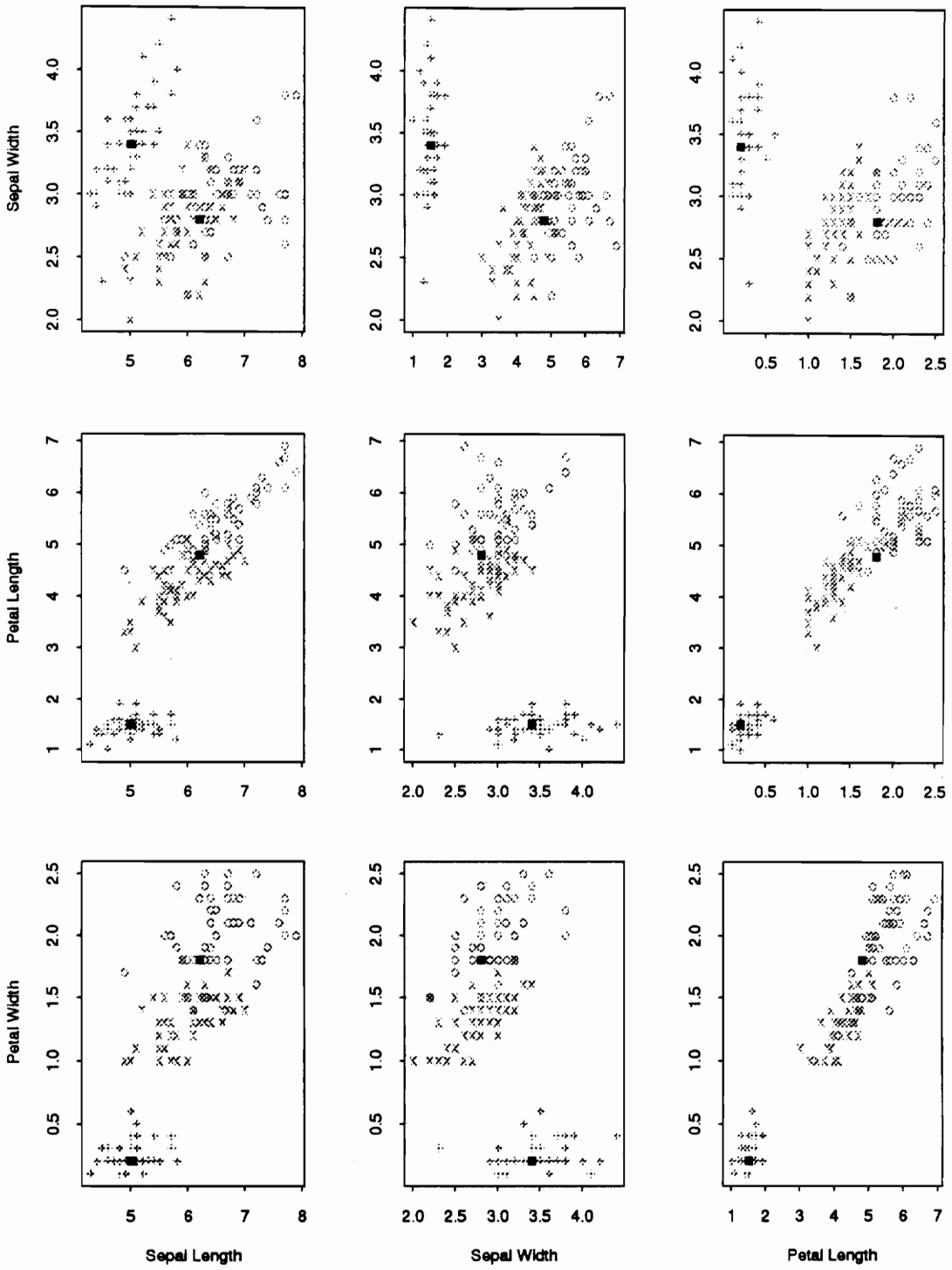


Figure 3-5 (B) : Positive weights by LSMDE for the Iris data (H=1.2). "■" denotes positive weights, "+" denotes Setosa, "x" denotes Vericolor and "o" denotes Virginica.Positive weights are 0.20626 for ID8, and 0.79374 for ID127.

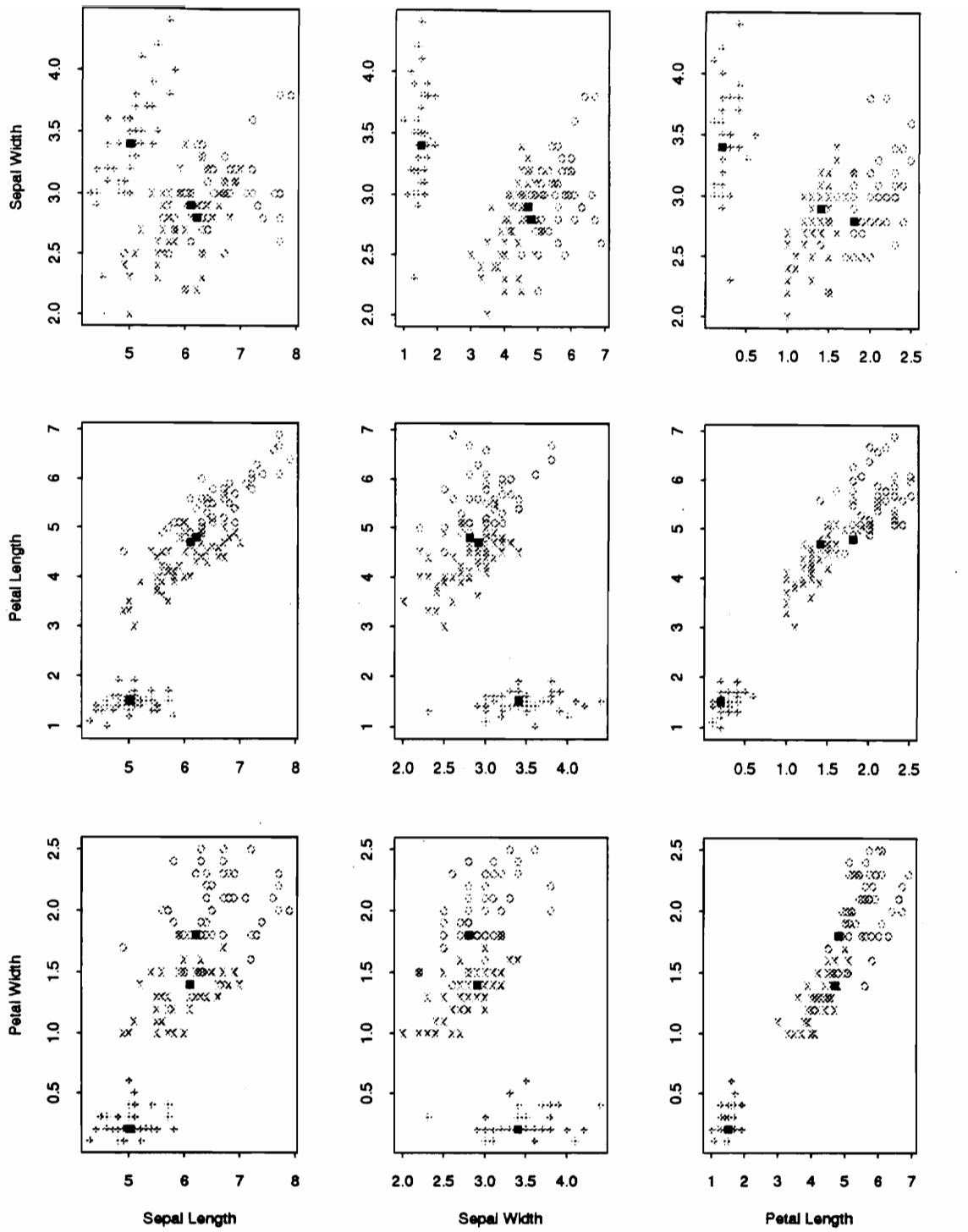


Figure 3-5 (C) : Positive weights by LSMDE for the Iris data (H=1.4). "■" denotes positive weights, "+" denotes Setosa, "x" denotes Vericolor and "o" denotes Virginica. Positive weights are 0.12770 for ID8, 0.11733 for ID64, and 0.75495 for ID127.

Chapter 4.

Simulation Study

Much attention has been paid to conditions under which the kernel estimate is, in various senses, a consistent estimate of the true density. The conditions for consistency are surprisingly mild, though the rate at which the estimated density converges to its true value can be extremely slow [Silverman (1986)]. Although very limited results for the unrestricted LSMDE (ULSMDE) are discussed in Appendix B, asymptotic properties of the LSMDE are unknown. In this chapter, we illustrate the asymptotic behavior of the LSMDE via Monte Carlo study. The main purpose of this study is to see how our new estimators behave as the sample size increases and whether it shows the characteristics of other available nonparametric density estimators. We have not attempted any automatic bandwidth selection. However, we suggest some guidelines and conjectures based on this study.

4.1 Asymptotic properties of kernel density estimator

Basic statistical properties of the kernel density estimator have already been discussed in Section 1.2.2. Based on the optimal smoothing parameter h^*, the optimal AMISE of the kernel density estimator is $O\left(n^{-2p/(2p+1)}\right)$, where p is the order of given kernel function. A common assumption made is that the smoothing parameter is a function of the sample size n satisfying

$$h \to 0 \text{ and } nh \to \infty, \text{ as } n \to \infty. \tag{4.1}$$

This condition implies that, while the smoothing parameter must get smaller as the sample size increases, it must not converge to zero as rapidly as $1/n$. The expected number of points in the sample falling in the interval $(x\text{-}h, x+h)$ must tend to infinity, however slowly, as n tends to infinity [Silverman p. 71 (1986)]. So as n increases the smoothing parameter should be decreased giving more resolution for large sample sizes. A general result for kernel density estimators is that as n increases h should decrease essentially as $O\left(n^{-1/(2p+1)}\right)$. The condition (4.1) with some additional assumptions on the underlying density function guarantees the consistency in MISE of the kernel density estimator.

A more general form of the AMISE is considered as the form of

$$AMISE(h) = \frac{a}{(d+r)nh^{(d+r)}} + \frac{b}{2p}h^{2p} \tag{4.2}$$

where (d, p, r) are positive integers, and a, b are positive constants that depend upon the density estimator and unknown density function [Scott p.58 (1992)]. The triple (d, p, r) refers to, respectively (1) the dimension of the data; (2) the order of the estimator's bias; (3) the order of the derivative being estimated. For the kernel estimator $(d, p, r) = (1, 2, 0)$ provided a kernel of order 2 is used and $a = \int K^2(y)dy$, $b = \left[\int y^2 K(y)dy\right]^2$. The smoothing parameter which minimizes (4.2) is given

$$h^* = \left(a/(nb)\right)^{1/(d+r+2p)}. \tag{4.3}$$

4.2 Simulation

4.2.1 Simulation setup

Simulation has been performed from (1) the standard normal distribution; (2) the mixture of two normal distributions, 0.5 N(-1.5,1) + 0.5 N(1.5,1). For each distribution sample sizes of 25, 50, 100, 150, 200, 250, 300, 350 and the smoothing parameter from 0.5 to 1.4 by the increment of 0.1 have been investigated. For each combination of the sample size and the smoothing parameter 100 replications have been simulated to obtain the ISE, the average of simulated ISE's ($\overline{ISE}$) for the estimate of MISE, and the number of positive weights. From each run, ISE's were calculated by Simpson's rule [Thisted (1989)] from the values of the LSMDE $\hat{f}_L(y)$ and $f(y)$ at 100 equally spaced points ranging from -5.0 to 5.0. The kernel function used is the standard normal distribution.

4.2.2 Results

We present simulation results mostly for n=100, 200, 300. Table 4-1 and 4-2 shows the mean, the standard deviation, and the average number of positive weights of 100 ISE's from each distribution. The shaded cell gives the smallest $\overline{ISE}$ (the estimate for the MISE) among different smoothing parameters for a given sample size. Log-log plots of $\overline{ISE}$'s are displayed in Figure 4-1 (a) and (b) for n = 100, 200, 300 and given smoothing parameters. First we can clearly see that the $\overline{ISE}$ decreases as n increases for

most smoothing parameters. The smoothing parameter which gives the smallest $\overline{ISE}$ is 1.0 except the case of n=25 (h =1.1) for both cases and we might use $\overline{ISE}(h = 1.0)$ as the estimate for the optimal MISE. From log-log plots of sample sizes versus $\overline{ISE}(h = 1.0)$'s (Figure 4-2 (a), (b)), we conjecture that the optimal MISE tends to decrease as n increases.

To see the asymptotic behavior of the LSMDE, we used nonlinear regression of the form (4.2) with a, b, and p as unknowns to fit $\overline{ISE}$'s. Estimated regression coefficients are listed in Table 4-3 for each case and fitted regression lines are displayed in Figure 4-1 (c) and (d). The optimal smoothing parameter was calculated from (4.3) and listed in Table 4-4, based on the fitted line, The calibrated optimal smoothing parameter seems to decreases, as n increases – for the example of $N(0,1)$, h^* decreases from 0.976449 for n=100 to 0.829283 for n =300. It is not clear that the simulation result supports the condition (4.1), since the nonlinear regression model (4.2) is a decreasing function in n and did not fit $\overline{ISE}$'s well in both cases. Quadratic regressions of $MISE = \beta_0 + \beta_1 h + \beta_2 h^2$ were fitted to calibrate h^* with inverse confidence limits (fiducial limits) for h^* based on the method by Williams (1959) [Draper and Smith (1981)]. Fitted lines and inverse confidence limits are displayed in Figure 4-1 (e) and (f) with Table 4-4. Estimated quadratic regression lines fitted better than lines based on (4.2) for each case and evidently h^* decreases as $n \to \infty$. As expected, the calibrated optimal bandwidth for the standard normal distribution example is slightly wider than for the mixture case in both regressions.

Average numbers of positive weights for n =300 are plotted in log-log axes (Figure 4-2 (c) and (d)). As h increases, the number of positive weights gets smaller in general. For $N(0,1)$, the average number of positive weights drops rapidly to 2 as h

reaches 1.1, then it remains stable suggesting oversmoothing. For the mixture example, the average number decreases up to 4 until h=1.2, then remains still afterwards. This plot might be useful in detecting the oversmoothing problem of the LSMDE. In Figure 4-3 (a) – (f), the histogram of actual numbers of positive weights for the case of n = 300, h = (0.7,1.0,1.4) is displayed. The mode of number of positive weights changes from 5 to 2 for N (0,1), and from 9 to 3 for the mixture as h increases. Therefore the number of positive weights seems a little larger than it should be even when the proper bandwidth is selected in both cases. We have two comments on this issue. First, the overestimation of the number of components can be relieved if we use higher cut-off guidelines for the positive weights – the cut-off point used to distinguish the positive weight from zero was 1e-12. Too small weights might be ignored in practice. Second, caution needs to be exercised when examining small weights, since, as we have seen from several examples, small weights usually locate around large weights. Small weights around large weights can be safely ignored in calculating the number of components and added to larger weights near them for simplification.

Table 4-1 : Simulation results for *N(0,1)*

For each cell, the mean, the standard deviation and the average number of positive weights of 100 ISE's are reported. Shaded cells give the smallest mean of generated samples among different bandwidths for a given sample size.

	N=25	N=50	N=100	N=150	N=200	N=250	N=300	N=350
H=0.5	0.033121	0.018099	0.010409	0.006529	0.005355	0.004634	0.003755	0.003301
	(0.02395)	(0.01197)	(0.00731)	(0.00434)	(0.00289)	(0.00274)	(0.00227)	(0.00205)
	5.32	6.44	7.29	7.77	8.03	8.39	8.32	9.02
H=0.6	0.026142	0.014619	0.008498	0.005286	0.004168	0.003655	0.002933	0.002628
	(0.02017)	(0.01085)	(0.00627)	(0.00398)	(0.00238)	(0.00243)	(0.00191)	(0.00178)
	4.51	5.45	6.10	6.67	6.69	6.92	7.21	7.25
H=0.7	0.021263	0.011718	0.006770	0.004389	0.003326	0.002937	0.002304	0.002109
	(0.01763)	(0.00983)	(0.00495)	(0.00367)	(0.00207)	(0.00215)	(0.00166)	(0.00159)
	4.11	4.59	5.03	5.61	5.66	5.69	6.00	6.01
H=0.8	0.017479	0.009312	0.005165	0.003578	0.002688	0.002387	0.001848	0.001726
	(0.01548)	(0.00905)	(0.00387)	(0.00317)	(0.00186)	(0.00188)	(0.00145)	(0.00141)
	3.56	3.95	4.23	4.67	4.85	4.92	5.21	5.05
H=0.9	0.013727	0.007311	0.003546	0.002787	0.002148	0.001771	0.001462	0.001315
	(0.01347)	(0.00859)	(0.00318)	(0.00271)	(0.00168)	(0.00155)	(0.00124)	(0.00115)
	3.01	3.45	3.47	3.94	4.10	3.98	4.15	4.13
H=1.0	0.011284	0.005940	0.002488	0.002091	0.001579	0.001078	0.001011	0.000852
	(0.01207)	(0.00761)	(0.00309)	(0.00240)	(0.00158)	(0.00132)	(0.00105)	(0.00095)
	2.58	2.71	2.63	2.80	2.89	2.76	2.90	2.89
H=1.1	0.011146	0.006346	0.003659	0.003238	0.002795	0.002457	0.002332	0.002304
	(0.01061)	(0.00624)	(0.00250)	(0.00181)	(0.00124)	(0.00096)	(0.00066)	(0.00068)
	2.23	2.11	1.97	1.93	1.96	1.88	1.97	1.89
H=1.2	0.013552	0.009470	0.007705	0.007393	0.007131	0.006858	0.006754	0.006758
	(0.00886)	(0.00499)	(0.00187)	(0.00131)	(0.00103)	(0.00075)	(0.00050)	(0.00056)
	2.00	1.83	1.68	1.73	1.84	1.77	1.81	1.79
H=1.3	0.018113	0.015015	0.013722	0.013448	0.013237	0.013006	0.012919	0.012922
	(0.00735)	(0.00400)	(0.00157)	(0.00106)	(0.00087)	(0.00061)	(0.00041)	(0.00046)
	1.75	1.66	1.71	1.71	1.73	1.64	1.75	1.68
H=1.4	0.024316	0.021827	0.020770	0.020530	0.020357	0.020159	0.020086	0.020090
	(0.00611)	(0.00341)	(0.00133)	(0.00087)	(0.00074)	(0.00049)	(0.00035)	(0.00039)
	1.68	1.61	1.73	1.57	1.75	1.65	1.64	1.70

Table 4-2 : Simulation results for *0.5 N(-1.5,1) + 0.5 N(1.5,1)*

For each cell, the mean, the standard deviation and the average number of positive weights of 100 ISE's are reported. Shaded cells give the smallest mean of generated samples among different bandwidths for a given sample size.

	N=25	N=50	N=100	N=150	N=200	N=250	N=300	N=350
H=0.5	0.034280	0.018845	0.010534	0.007010	0.006529	0.004808	0.004145	0.003808
	(0.01759)	(0.00969)	(0.00479)	(0.00301)	(0.00262)	(0.00217)	(0.00182)	(0.00166)
	7.64	9.06	10.4	11.13	11.87	12.4	12.18	12.71
H=0.6	0.028118	0.015312	0.008685	0.005810	0.005286	0.003918	0.003331	0.003153
	(0.01626)	(0.00801)	(0.00434)	(0.00287)	(0.00231)	(0.00191)	(0.00151)	(0.00163)
	6.63	7.73	8.95	9.57	9.86	10.43	10.45	10.58
H=0.7	0.023770	0.012556	0.007374	0.005007	0.004451	0.003280	0.002818	0.002654
	(0.01573)	(0.00693)	(0.00399)	(0.00272)	(0.00213)	(0.00176)	(0.00141)	(0.00150)
	5.87	6.95	7.83	8.23	8.52	8.79	8.99	8.95
H=0.8	0.020536	0.010502	0.006230	0.004226	0.003857	0.002755	0.002442	0.002213
	(0.01544)	(0.00650)	(0.00347)	(0.00258)	(0.00198)	(0.00157)	(0.00135)	(0.00129)
	5.24	5.93	6.71	6.98	7.41	7.62	7.69	7.74
H=0.9	0.018065	0.009083	0.005255	0.003496	0.003285	0.002307	0.002092	0.001810
	(0.01508)	(0.00635)	(0.00318)	(0.00243)	(0.00187)	(0.00148)	(0.00131)	(0.00113)
	4.63	5.21	5.77	5.83	6.3	6.25	6.62	6.41
H=1.0	0.016537	0.008141	0.004627	0.003006	0.002890	0.001949	0.001736	0.001505
	(0.01450)	(0.00600)	(0.00308)	(0.00240)	(0.00186)	(0.00140)	(0.00128)	(0.00109)
	4.18	4.46	4.89	4.83	5.05	4.91	4.75	5.18
H=1.1	0.016020	0.008152	0.004788	0.003334	0.003148	0.002328	0.002157	0.001874
	(0.01388)	(0.00564)	(0.00280)	(0.00221)	(0.00167)	(0.00128)	(0.00117)	(0.00090)
	3.85	3.97	3.94	3.92	4.18	3.98	3.92	4.05
H=1.2	0.016305	0.008971	0.005826	0.004589	0.004299	0.003690	0.003556	0.003254
	(0.01329)	(0.00522)	(0.00250)	(0.00198)	(0.00146)	(0.00118)	(0.00106)	(0.00075)
	3.52	3.65	3.72	3.6	3.62	3.6	3.51	3.56
H=1.3	0.016855	0.010272	0.007307	0.006233	0.005936	0.005456	0.005312	0.005027
	(0.01236)	(0.00486)	(0.00225)	(0.00186)	(0.00136)	(0.00111)	(0.00099)	(0.00068)
	3.3	3.37	3.39	3.38	3.41	3.39	3.44	3.44
H=1.4	0.017503	0.011644	0.008866	0.007892	0.007632	0.007194	0.007066	0.006782
	(0.01116)	(0.00462)	(0.00210)	(0.00177)	(0.00132)	(0.00104)	(0.00095)	(0.00063)
	3.18	3.42	3.36	3.33	3.47	3.32	3.37	3.27

Table 4-3 : Nonlinear regression coefficients

	N(0,1)			*0.5 N(-1.5,1) + 0.5 N(1.5,1)*		
Sample size	*a*	*b*	*p*	*a*	*b*	*p*
100	0.444820	0.005827	5.166099	0.506280	0.002468	4.431453
200	0.441285	0.008592	4.463555	0.618008	0.002841	4.176666
300	0.442582	0.009107	4.361702	0.576794	0.003229	3.995430

Table 4-4 : Optimal smoothing parameters and minimized MISE from regression fits.
[LB(h^*), UB(h^*)] : 95 % lower bound and upper bound for calibration at $h=h^*$ from quadratic regression fit.

(A) *N(0,1)*

	Nonlinear Regression		Quadratic Regression			
Sample size	h^*	MISE(h^*)	h^*	MISE(h^*)	LB(h^*)	UB(h^*)
100	0.976449	0.004996	0.888033	0.002830	0.835784	0.940282
200	0.872015	0.002814	0.831968	0.000899	0.776284	0.887653
300	0.829283	0.001983	0.815903	0.000088	0.758054	0.873752

(B) *0.5 N(-1.5,1) + 0.5 N(1.5,1)*

	Nonlinear Regression		Quadratic Regression			
Sample size	h^*	MISE(h^*)	h^*	MISE(h^*)	LB(h^*)	UB(h^*)
100	1.075570	0.005238	0.997737	0.004939	0.971606	1.023868
200	1.009028	0.003429	0.933068	0.003071	0.899192	0.966945
300	0.943968	0.002292	0.880484	0.001842	0.837333	0.923636

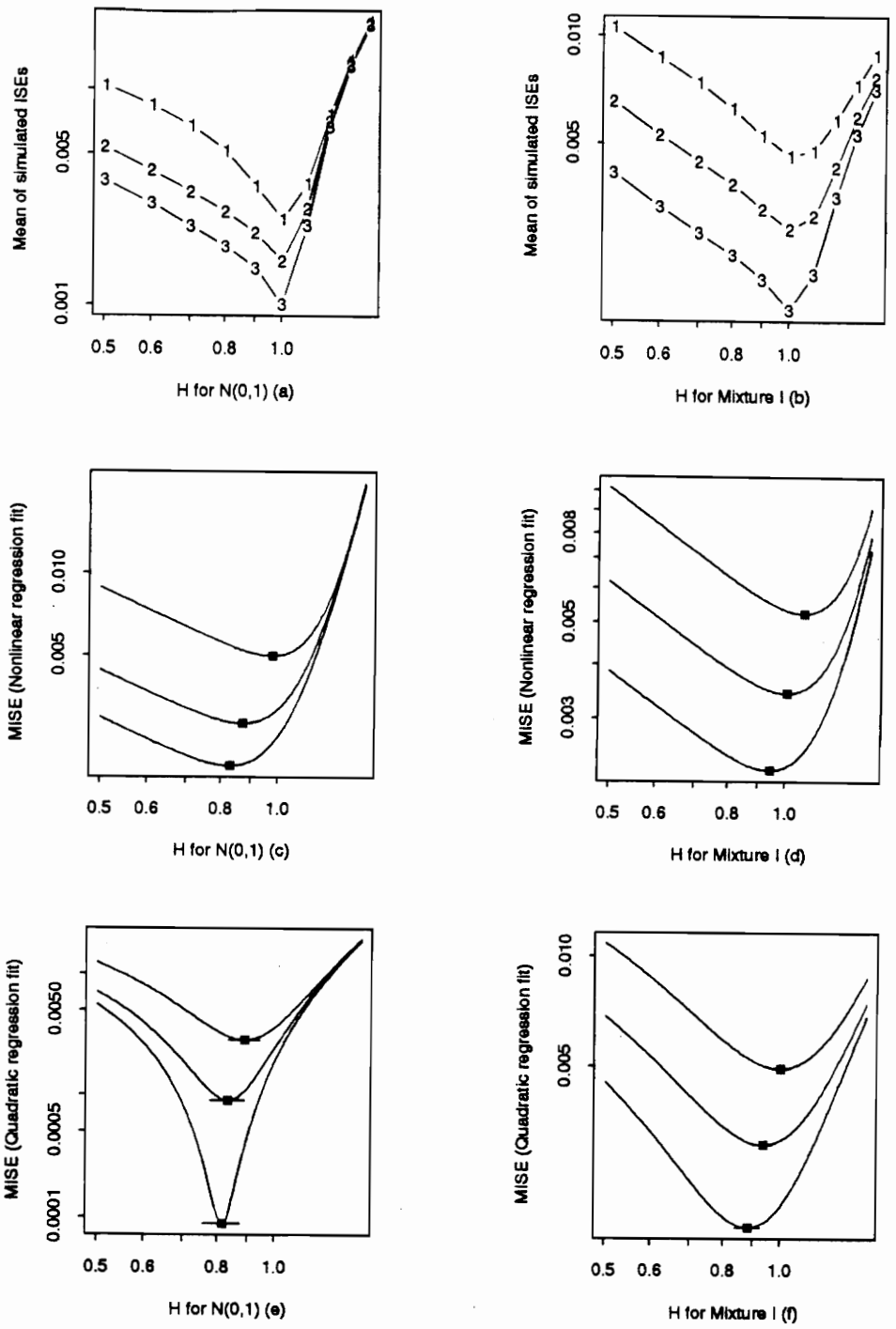


Figure 4-1 : (a), (b) Bandwidth versus mean of simulated ISE's in log-log axes (1 for n=100, 2 for n=200, 3 for n=300). (c), (d) Nonlinear regression fit of AMISE in log-log axes. (e), (f) Quadratic regression fit with inverse confidence limits in log-log axes. "■" denotes the calibrated minimum for optimal bandwidth.

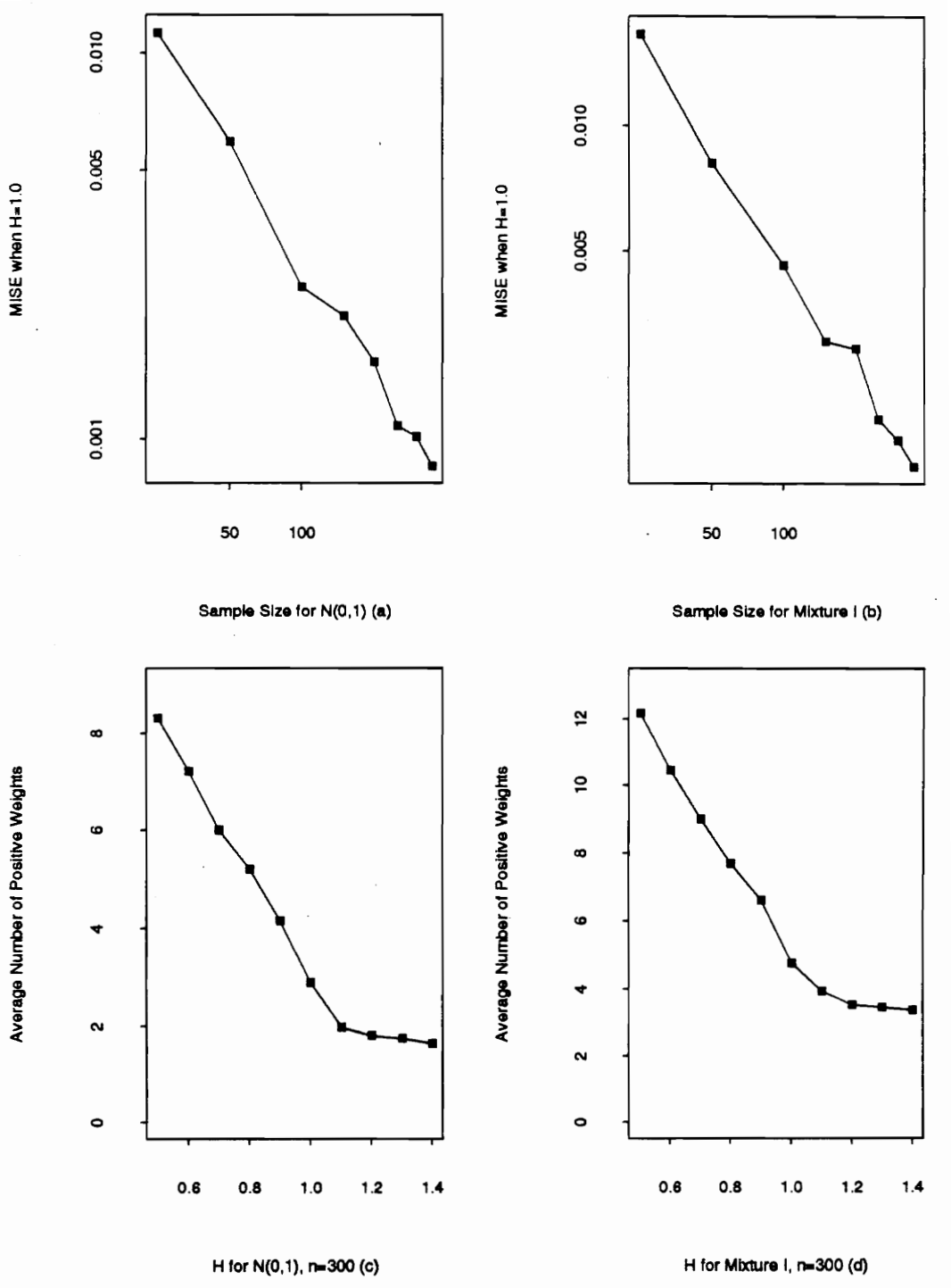


Figure 4-2 : (a), (b) Sample size versus simulated MISE at h=1.0 in log-log axes. (c), (d) Bandwidth versus average number of positive weights for n=300 in log-log axes.

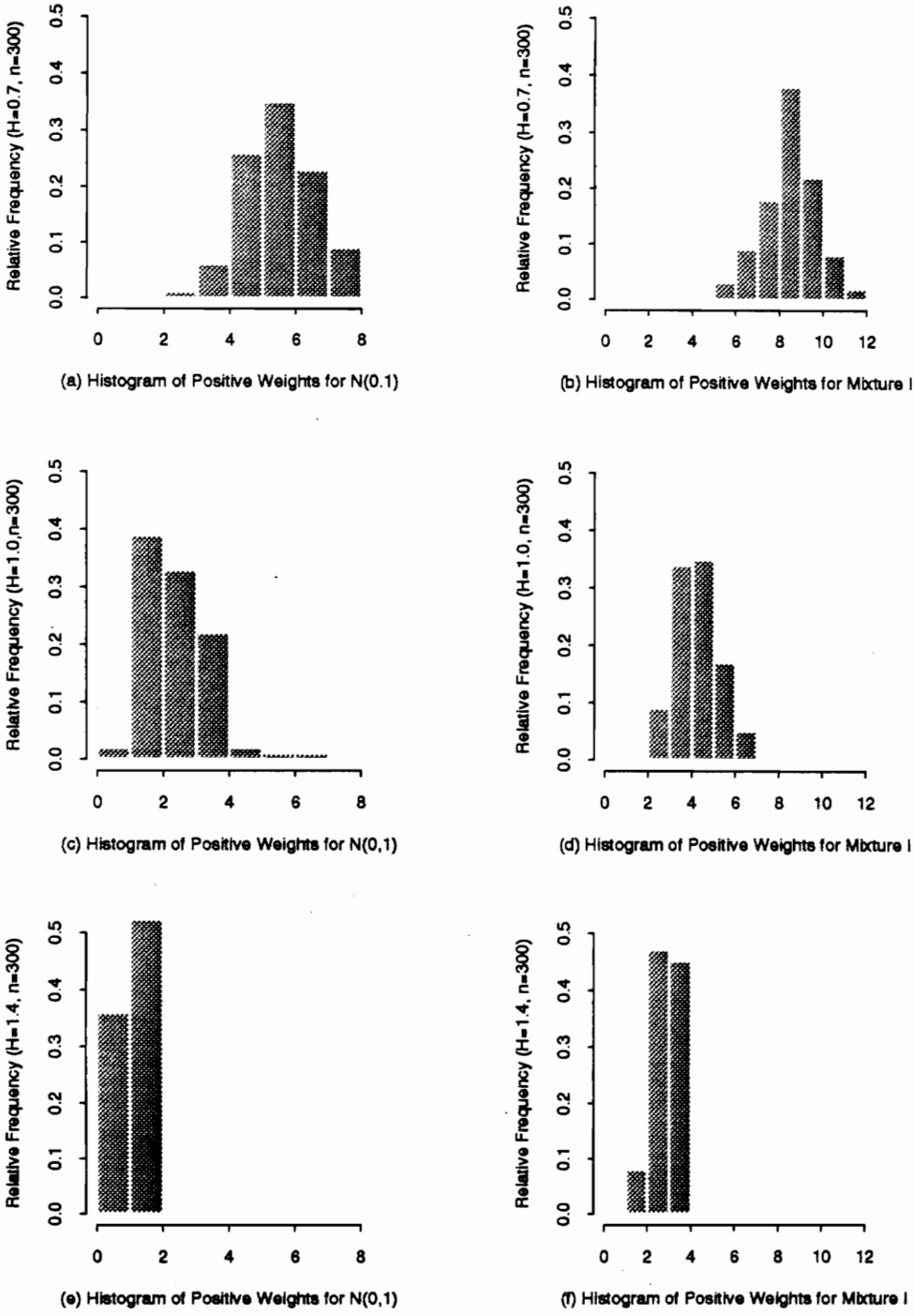


Figure 4-3 : Histogram in relative frequency of positive weights for h=1.0, n=300.

Chapter 5

Summary and Discussion

Many examples shown seem to indicate that the LSMDE performs well for several points of view. We have developed the LSMDE in an attempt to combine the kernel density estimate and the finite mixture distribution approach. The LSMDE was successful in estimating density functions by identifying (1) the number of mixture components (2) the mixing weights. As a nonparametric density estimation technique, the LSMDE provides estimates very similar to the ordinary kernel estimator and more informative by suggesting possible structure of the underlying density functions. The LSMDE behaves well enough to locate the mode of the underlying density functions indirectly with a quite small number of mixture components decomposing the true density functions. Although we were not able to show common asymptotic properties of the LSMDE, we hope that its asymptotic results are as good as the kernel estimator, since the LSMDE is based on weights optimized for a data driven estimate of the MISE.

The LSMDE has its advantage over the ordinary kernel estimator in that it can be interpreted as a mixture distribution. It, furthermore, can be considered as an estimation method for finite mixture models especially when the number of components is unknown. For normal mixture models with common variances, the LSMDE can be useful in identifying the number of components and the location parameter. It appears that the application for exponential mixtures is more justified. While maximum likelihood estimators by the EM algorithm seem to outperform the LSMDE, the LSMDE has the advantage that the number of components need not be specified. Application to higher dimension is fairly straightforward and insightful.

We mention several problems we have encountered while developing the LSMDE. As usual, when applied as a nonparametric density estimator, the problem of optimal bandwidth appears. We have not attempted to apply any of automatic selection methods, since (1) it costs too much to use a cross validation method, (2) there is little consensus on universally applicable methods, and (3) an exploratory approach to choosing a smoothing parameter is advocated by many statisticians. Therefore we have tried to select the bandwidth based on changes in optimized positive weights while trying wide ranges of bandwidths. We have seen that the number of positive weights in the LSMDE does not always decrease as h increases. In some cases, the number of positive weights stays the same for a quite broad range of h. This might be one guideline for choosing a smoothing parameter as well. However some of the nonzero weights are negligible compared to the significant weights. If we want to use the number of positive weights as an indication of the number of components in mixture and the density estimates, we should be able to determine which weights are significant and which weights should be considered negligible. Lack of asymptotic results makes us wonder how the LSMDE will behave for many diverse practical situations.

A disappointing aspect of the LSMDE is the computational speed of the optimal weight algorithm. The computing time required for common quadratic programming algorithms increases exponentially as the problem size increases. This disadvantage makes the LSMDE almost impossible to be applied as an effective exploratory tool. Since we have to solve a quadratic programming of size n, the application of the ordinary LSMDE should be avoided for large data sets. We recommend the discretized version for one or two dimensional cases. As we mentioned, the discretized LSMDE has the advantage of numerical stability in addition to the reduction of problem sizes. However,

the discretized version cannot be applied for higher dimension data sets, since the number of regular mesh points increases rapidly as the dimension goes up. Although we introduced an alternative method for quadratic programming problems, algorithms based on the constrained least squares are not recommended. The FORTRAN routine QLD.F which has been used throughout our examples needs to be modified further to take advantage of the special structure of constraints.

Finally, a proof of consistency will greatly improve the value of the LSMDE in practice.

Appendix : Computational details and figures

A. Convex quadratic programming

A.1 Convex programming

A convex programming problem is an optimization problem of the form

$$\begin{aligned} &\min_{\underline{x}\in R^n} \quad F(\underline{x}) \\ \text{s.t} \quad &c_i(\underline{x}) = b_i \quad i = 1,\ldots,m' \\ &c_i(\underline{x}) \geq b_i \quad i = m'+1,\ldots,m \end{aligned} \tag{A.1}$$

in which $F(\underline{x})$ is convex, the equality constraints are linear, and inequality constraints are concave. A fundamental property of a convex programming problem is that any local minimum is a global minimum; furthermore, if the Hessian matrix of $F(\underline{x})$ is positive definite, the solution is unique [Gill, Murray and Wright (1981) p. 257, Fletcher (1987) p. 216]. Examples are the linear programming and the quadratic programming with positive semi-definite Hessian matrix.

A.2 QPROG and QLD.F

The routine QPROG is based on M. J. D. Powell's implementation of the Goldfarb and Idnani (1983) dual quadratic programming algorithm for convex QP problems subject to general linear equality/inequality constraints of the form

$$\min_{\underline{a}\in R^n} \quad \underline{a}^T \underline{d} \quad + \frac{1}{2}\underline{a}^T C \underline{a}$$

$$\text{s.t.} \quad A_1 \underline{a} = \underline{b}_1,$$
$$A_2 \underline{a} \geq \underline{b}_2 \qquad \text{(A.2)}$$

given the vectors $\underline{b}_1$, $\underline{b}_2$, and $\underline{d}$ and the matrices C, A_1 and A_2. C is required to be positive definite. In this case, a unique a solves the problem or constraints are inconsistent. If C is not positive definite, a positive definite perturbation of C is used in place of C.

QLD.F is a modification of public domain FORTRAN routines due to M. J. D. Powell (1983) by K. Schittkowski for solving convex quadratic programming (available at laplace.stat.ucla.edu). Users can additionally specify the lower and upper bound of solutions and use factorized input of C [See Appendix D].

A.3 Relationship between QP and least squares

The convex quadratic programming problem is closely related to the problem of the least squares estimation in linear regression. The difference lies in the fact that least squares estimation usually deals with over-determined systems.

We illustrate how the constrained least squares estimation can be used to solve the quadratic programming problem. For computational aspects of constrained least squares, see Lawson and Hanson (1974).

Consider the linearly constrained linear least squares problem ;

$$\underline{y} = X \underline{a} + \underline{\varepsilon}$$
$$\text{s.t.} \quad A_1 \underline{a} = \underline{b}_1, \qquad A_2 \underline{a} \geq \underline{b}_2 \qquad \text{(A.3)}$$

where X is an (n x p) fixed known matrix with full column rank, $n \geq p$, $\underline{y}$ is an (n x 1) known vector, $\underline{a}$ is a (p x 1) unknown vector, $\underline{\varepsilon}$ is an (n x 1) vector of random error, and A_1, A_2, b_1, b_2 are defined as in (A.2). To obtain the least squares estimator, we wish to minimize the sum of squares of residuals, $\underline{e}^T\underline{e} = \left\|\underline{y} - X\underline{a}\right\|_2^2 = (\underline{y} - X\underline{a})^T(\underline{y} - X\underline{a})$. This is equivalent to

$$\min_{\underline{a}} \underline{a}^T X^T X \underline{a} - 2\underline{a}^T X^T \underline{y}$$

$$\text{s.t.} \quad A_1\underline{a} = \underline{b}_1, \qquad A_2\underline{a} \geq \underline{b}_2. \tag{A.4}$$

It is obvious that there exists the exact correspondence between (A.2) and (A.3), since

$$C = (\sqrt{2}X)^T(\sqrt{2}X),$$

$$\underline{d} = -2X^T\underline{y},$$

and p becomes n. Since X is assumed to have a full column rank of n, C becomes positive definite, and the property of a convex programming discussed in A.1 holds for (A.4). Now suppose we want to solve (A.2) by (A.3).

STEP 1 : Compute the Cholesky decomposition of C [Golub and Van Loan (1983)].

$C = L L^T$, where L is an (n x n) lower triangular matrix.

STEP 2 : Compute

$$X = \frac{1}{\sqrt{2}}L^T, \text{ and } \underline{y} = -\frac{1}{\sqrt{2}}L^{-1}\underline{d}.$$

STEP 3 : Solve (A.3) with $\underline{y}$ and X using conventional statistical packages.

Discussion based on QR decompositions can be found in Bartels, Golub, and Saunders (1970).

Most major conventional statistical packages such as SAS, SPSS, and BMDP and commercial mathematical libraries like IMSL, NAG have routines for the (linearly) constrained least squares problem. Matrix oriented languages such as SAS/IML and Matlab (optimization toolbox) have a function for QP problems. Also WNNLS, a public domain FORTRAN subroutine for a linearly constrained least squares problem with equality and nonnegativity constraints based on Lawson and Hanson (1974) is available at statlib@lib.stat.cmu.edu. (/cmlib).

B. Variants of the LSMDE

As we define in Chapter 2, the LSMDE is based on the constrained – both linear equality and inequality – optimization on the weights for each observation. One might consider two possible variants of the LSMDE, one without the constraint, the other only with the equality constraint. Although we can save computational effort considerably by relaxing the weight condition, the resulting density estimate can take on negative values and no longer has the conceptual advantage of the LSMDE.

B.1 The unconstrained LSMDE

B.1.1 Definition and estimation

Definition : Unconstrained LSMDE of $f(y)$ (ULSMDE)

$\hat{f}_u(y)$ is defined as the unconstrained LSMDE of $f(y)$

if given an i.i.d random sample of size n, $\{X_i\}$ i=1,..,n , the kernel function $K(u)$, and the fixed smoothing parameter h ,

$$\hat{f}_u(y) = \sum_{i=1}^{n} a_i K_h(y - x_i) \tag{B.1}$$

where $\{a_i\}$, i=1,..,n minimize the objective function,

$$\hat{Q}_u[\hat{f}(y)] = -\frac{2}{n}\sum_{i=1}^{n} \hat{f}(X_i) + \int \hat{f}^2(y)dy = -2\,\underline{a}^T \underline{d} + \underline{a}^T C \underline{a} \tag{B.2}$$

where

$$\underline{d} = \begin{pmatrix} \frac{1}{n}\sum_{i=1}^{n} K_h(x_1 - x_i) \\ \vdots \\ \frac{1}{n}\sum_{i=1}^{n} K_h(x_n - x_i) \end{pmatrix} = \begin{pmatrix} \hat{f}_K(x_1) \\ \vdots \\ \hat{f}_K(x_n) \end{pmatrix},$$

$$C = \left\{ K_h^*(x_i - x_j) \right\}_{i=1,\ j=1}^{n,\ n},$$

and $\hat{f}_K(x_i)$ denotes the ordinary kernel density estimator at $X = x_i$ with the bandwidth h. Note that $\underline{d}$ and C are defined differently from the LSMDE (2.8) by constant for convenience.

Differentiation with respect to $\underline{a}$ of $\hat{Q}_u[\hat{f}(y)]$ leads to the stationary condition

$$C\underline{a} = \underline{d} \quad \text{or} \quad \hat{\underline{a}}_u = C^{-1}\underline{d} \quad \text{provided that } C \text{ is invertible.} \tag{B.3}$$

Since C is proven to be positive definite by proposition 2.2, $\hat{f}_u(y)$ always exists and is unique for a given random sample and bandwidth. Actual density estimate can be carried out much easier than the ordinary LSMDE by solving the system of linear equations.

B.1.2 Properties

From the stationary condition (B.3), we observe that the optimized weight, $\underline{\hat{a}}_u$ is a linear combination of the ordinary kernel density estimator.

Proposition B.1.2.1 : Asymptotic properties of the unconstrained LSMDE

For a nonnegative univariate kernel function and bandwidth h_0,

1. $Bias_{h=h_0}[\hat{f}_u(x_i)] \le Bias_{h=h_0}[\hat{f}_K(x_i)]$
2. $Var_{h=h_0}[\hat{f}_u(x_i)] \ge Var_{h=h_0}[\hat{f}_K(x_i)]$

Proof :

Note that $E[C\underline{\hat{a}}_u] = E[\underline{d}]$ is the expectation of the ordinary kernel density estimators for a given random sample and bandwidth.

Since

$$C\underline{\hat{a}}_u = \left\{ \sum_{j=1}^{n} \hat{a}_j K_h^*(x_i - x_j) \right\}_{i=1}^{n},$$

for $X = x_i$, from (1.3), and (1.4)

$$E\left[\sum_{j=1}^{n} \hat{a}_j K_h^*(x_i - x_j) \right] = E\left[\frac{1}{n} \sum_{j=1}^{n} K_h(x_i - x_j) \right] \approx f(x_i) + \frac{1}{2}\sigma_K^2 h^2 f''(x_i),$$

$$Var\left[\sum_{j=1}^{n}\hat{a}_j K_h^*(x_i - x_j)\right] = Var\left[\frac{1}{n}\sum_{j=1}^{n}K_h(x_i - x_j)\right] \approx \frac{1}{nh}f(x_i)R(K).$$

Suppose the bandwidth of the convolution kernel $K_h^*(u)$ is h_1 provided that the bandwidth for the unconstrained LSMDE is h_0. Then $h_1 \geq h_0$, since $K_{h=h_1}^*(u)$ is the convolution of two $K_{h=h_0}(u)$'s. This leads to the following inequalities:

$$Bias\left[\frac{1}{n}\sum_{j=1}^{n}K_{h=h_0}(x_i - x_j)\right] = Bias\left[\sum_{j=1}^{n}\hat{a}_j K_{h=h_1}^*(x_i - x_j)\right] \geq Bias\left[\sum_{j=1}^{n}\hat{a}_j K_{h=h_0}(x_i - x_j)\right],$$

$$Var\left[\frac{1}{n}\sum_{j=1}^{n}K_{h=h_0}(x_i - x_j)\right] = Var\left[\sum_{j=1}^{n}\hat{a}_j K_{h=h_1}^*(x_i - x_j)\right] \leq Var\left[\sum_{j=1}^{n}\hat{a}_j K_{h=h_0}(x_i - x_j)\right].$$

Hence, the unconstrained LSMDE has smaller bias but larger variance than the ordinary LSMDE when the estimates are evaluated at data points. ■

Proposition B.1.2 : Asymptotic Properties of the unconstrained Normal LSMDE

For a Normal kernel function and bandwidth h,

as $h \to 0$ and $nh \to \infty$,

1. $Bias[\hat{f}_u(x_i)] = \frac{1}{4}\sigma_K^2 h^2 f''(x_i) + O(h^4)$
2. $Var[\hat{f}_u(x_i)] = \frac{\sqrt{2}}{nh}f(x_i)R(K) + O\left(\frac{h}{n}\right)$
3. $MSE[\hat{f}_u(x_i)] = \frac{\sqrt{2}}{nh}f(x_i)R(K) + \frac{1}{16}\sigma_K^4 h^4[f''(x_i)]^2 + O(h^4) + O\left(\frac{h}{n}\right)$
4. $h_{MSE}^* = \sqrt{2}\left[\frac{f(x_i)R(K)}{\sigma_K^4[f''(x_i)]^2}\right]^{1/5} n^{-1/5} = \sqrt{2}\,h^*$
5. $AMISE[\hat{f}_u(x_i)] = \sqrt{2}\frac{R(K)}{nh} + \frac{1}{16}\sigma_K^4 h^4 R(f'')$

6. $h^{*}_{MISE} = \sqrt{2}\left[\frac{R(K)}{\sigma_K^4 R(f'')}\right]^{1/5} n^{-1/5} = \sqrt{2}\, h^{***}$

7. $AMISE^{*}[\hat{f}_u(x_i)] = \frac{5}{4}[\sigma_K R(K)]^{4/5} R(f'')^{1/5} n^{-4/5}$

8. the ratio of IV to ISB = 1:4.

where $R(f) = \int f^2(y)dy$, $\sigma_K^2 = \int y^2 K(y)dy$, h^{*} and h^{***} is the bandwidth which minimizes AMSE and AMISE of the ordinary kernel density estimates respectively.

Proof : Consider the ordinary kernel density estimate with bandwidth $h/\sqrt{2}$. Then the convolution of two standard normal kernels becomes a normal kernel with bandwidth h. From proposition B.1,

$$E\left[\sum_{j=1}^{n} \hat{a}_j K_h(x_i - x_j)\right] = E\left[\frac{1}{n}\sum_{j=1}^{n} K_{h/\sqrt{2}}(x_i - x_j)\right].$$

Therefore, all properties of the unconstrained LSMDE correspond to those of the ordinary kernel density estimator with bandwidth $h/\sqrt{2}$. Details of each property follow from Scott (1992), Härdle (1991), or Silverman (1985). ■

B.2 The LSMDE only with the equality constraint

Another alternative of the LSMDE is one only with a linear equality constraint, $\underline{a}^T \underline{1} = 1$. The elimination of positivity condition on the weights could result in negative density estimates.

Definition : Linear-equality constrained LSMDE of $f(y)$ (ELSMDE)

$\hat{f}_e(y)$ is defined as the unconstrained LSMDE of $f(y)$

if given an i.i.d random sample of size n, $\{X_i\}$ i=1,..,n , the kernel function $K(u)$, and the fixed smoothing parameter h ,

$$\hat{f}_e(y) = \sum_{i=1}^{n} a_i K_h(y - x_i) \tag{B.4}$$

where $\{a_i\}$, i=1,..,n minimize the objective function,

$$\hat{Q}_e[\hat{f}(y)] = -\frac{2}{n}\sum_{i=1}^{n} \hat{f}(X_i) + \int \hat{f}^2(y)dy = -2\,\underline{a}^T \underline{d} + \underline{a}^T C \underline{a}$$

$$\text{s.t.} \quad \underline{a}^T \underline{1} = 1 \tag{B.5}$$

where $\underline{d}$, C and $\hat{f}_K(x_i)$ are defined as in (B.2).

By introducing Lagrange multiplier, the problem reduces to the unconstrained minimization as follows :

$$\min_{\underline{a},\lambda} H(\underline{a},\lambda) = -2\underline{d}^T \underline{a} + \underline{a}^T C \underline{a} + 2\lambda\,(\underline{1}^T \underline{a} - 1). \tag{B.6}$$

Differentiation with respect to $\underline{a}$ and λ gives

$$-\underline{d}^T \underline{a} + C\underline{a} + \lambda \underline{1} = \underline{0}\,, \qquad \underline{1}^T \underline{a} - 1 = 0. \tag{B.7}$$

Therefore,

$$\underline{a} = C^{-1}\underline{d} - \lambda\, C^{-1}\underline{1}. \tag{B.8}$$

By multiplying both sides by $\underline{1}^T$, we get

$$\lambda = \frac{\underline{1}^T C^{-1} \underline{d} - 1}{\underline{1}^T C^{-1} \underline{1}} \tag{B.9}$$

Finally, we can write down the solution explicitly as

$$\hat{\underline{a}}_e = C^{-1} \underline{d} - \left(\frac{\underline{1}^T C^{-1} \underline{d} - 1}{\underline{1}^T C^{-1} \underline{1}} \right) \underline{1} \tag{B.10}$$

The existence and uniqueness of $\hat{f}_e(y)$ follow from proposition 2.1, and 2.2. Contrary to the case of the ULSMDE, C^{-1} needs to be explicitly calculated.

Algorithms based on the linear least squares with one equality constraint can be found in Golub and Van Loan (1983).

B.3 Examples

A S-plus routine for the unconstrained and linearly constrained LSMDE routine, dlsm1() is implemented. Due to the computational singularity, the discretized version of the unconstrained LSMDE is considered. Even if the discretization of the problem alleviates the near-singularity, there still exists possibility of computational singularity. This is because common kernel functions such as the standard normal kernel become almost flat in their tails, unless the support of the kernel function is finite. Suppose observations are sorted. Let $\Delta_{ij} = x_i - x_j$. As Δ_{ij} becomes larger – the more separate two data points, the (i, j)-th element of C, $K_h^*(\Delta_{ij})$ becomes smaller. Therefore elements on the upper right or lower left corner of C tend to be extremely small when the normal kernel is used. Those extremely small numbers can easily cause overflow error when C is inverted. To reduce this difficulty, the elements of C are truncated at a proper decimal point when we calculate the examples.

Same data sets used in example 2.3.1 (the standard normal distribution) and 2.3.2 (the mixture of 2 normal densities) were considered. First there seems to be little difference between the ULSMDE and ELSMDE in both examples. Resulting estimates for h = 0.5, and 1.0 look similar to the ordinary LSMDE except that the ULSMDE and ELSMDE take negative values for some ranges [Figure B-1, B-2]. On the other hand, when h =1.6, while the ordinary LSMDE becomes oversmoothed rapidly, the ULSMDE and ELSMDE provide less smoothed estimates especially in the example of the mixture showing two bumps clearly. In the $N(0,1)$ example (h = 1.5), we note that a large portion of the tails becomes negative rather than oversmoothed as in Figure 2-1-1 (B).

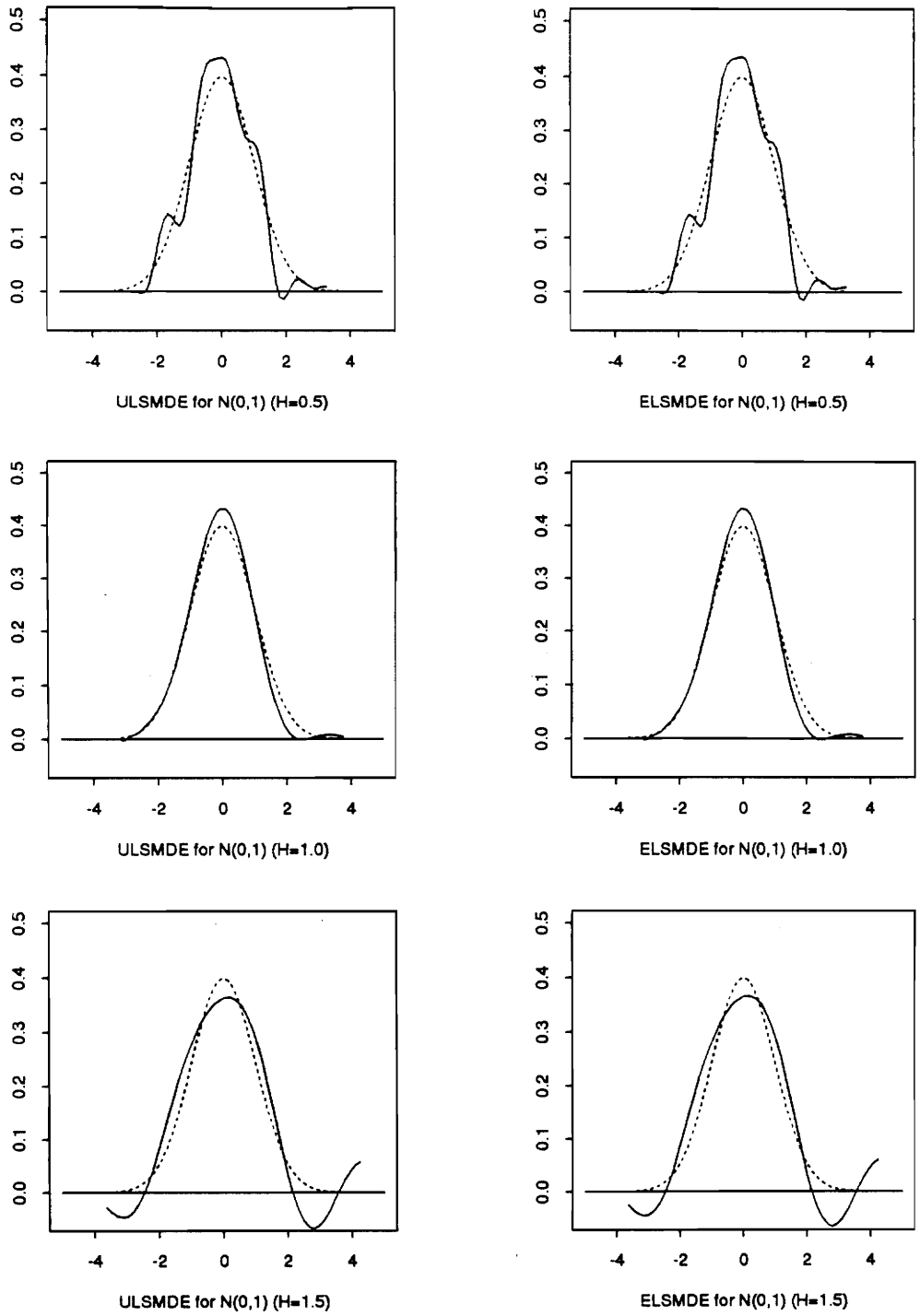


Figure A-1 : ULSMDE and ELSMDE for $N(0,1)$. The dotted line is $N(0,1)$.

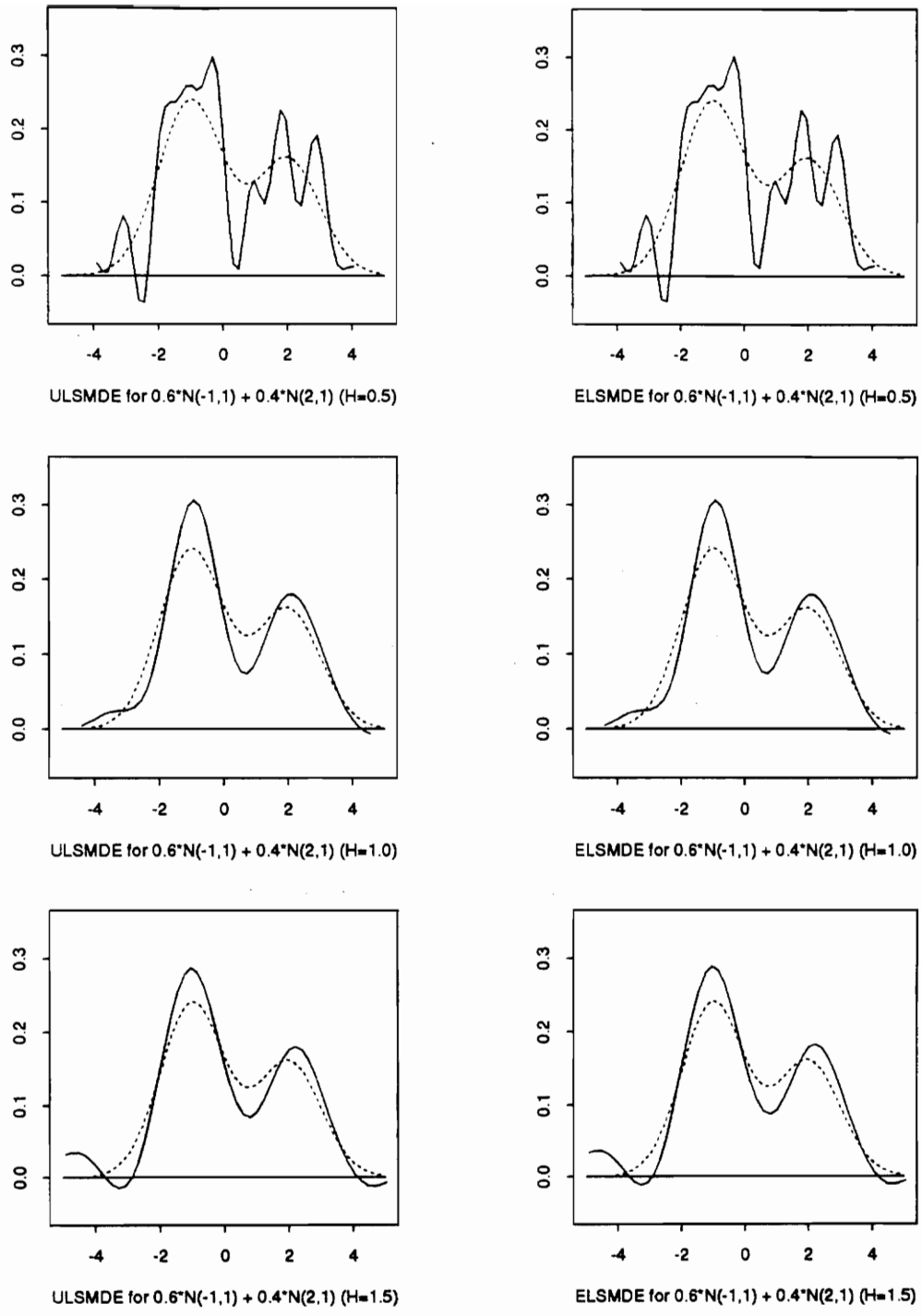


Figure A-2 : ULSMDE and ELSMDE for 0.6 $N(-1,1)$ + 0.4 $N(2,1)$. The dotted line is the true density.

C. Convolution of two extreme value distributions

Let $\Delta_{ij} = x_i - x_j$, and $K_h(u) = e^u \exp(-e^u)$.

Then the convolution kernel,

$$K_h^*(\Delta_{ij}) = \int K_h(y - x_i) K_h(y - x_j) dy$$

$$= \int \exp(y - x_i) \exp(-e^{(y-x_i)}) \exp(y - x_j) \exp(-e^{(y-x_i)}) dy$$

$$= e^{-(x_i + x_j)} \int e^{2y} \exp[-e^y (e^{-x_i} + e^{-x_j})] dy. \qquad \text{(C.1)}$$

Let $z = e^y$ so that $y = \ln(z), dy = \frac{1}{z} dz$, then

$$= e^{-(x_i + x_j)} \int z \exp[-z(e^{-x_i} + e^{-x_j})] dz$$

$$= \frac{e^{-(x_i + x_j)}}{(e^{-x_i} + e^{-x_j})} \int z(e^{-x_i} + e^{-x_j}) \exp[-z(e^{-x_i} + e^{-x_j})] dz. \qquad \text{(C.2)}$$

Since the integration in (C.2) is the expectation of exponential distribution with rate parameter $(e^{-x_i} + e^{-x_j})$, (C.2) becomes

$$= \frac{e^{-(x_i + x_j)}}{(e^{-x_i} + e^{-x_j})^2}$$

$$= \frac{\exp(\Delta_{ij})}{(1 + \exp(\Delta_{ij}))^2},$$

which is the standard logistic density.

D. Program Listings

```
#-------------------------------------------------------------------------+
# lsm    Calculating the least square mixture decomposition               |
#                                                                         |
# ARGUMENT                                                                |
#     x: vector/matrix of observations from f(x).                         |
#    ng: number of equally spaced grid points where to estimate f(x).     |
#     h: bandwidth. Default is the reference value for standard normal    |
#        density 0.96*n^(-1/id+4). (see Silverman(1986, p87)),where id    |
#        is the dimension of x                                            |
#  from: starting point of ng equally spaced grid points.                 |
#        Default is the min(xi)-h*0.75.                                   |
#    to: end points of ng equally spaced grid points.                     |
#        Default is the max(xi)+h*0.75.                                   |
#  dens: If density==T, actual density estimates are returned.            |
#        Otherwise only the optimized weights are returned                |
#   eps: epsilon for QL0001 for machine precision. It is also used to     |
#        weight selection parameter (as zero). Default is 1e-12.          |
#                                                                         |
# VALUE                                                                   |
#        List of three different types for suitable plot.                 |
#        If dimension is greater than 3 or density=F, only the weight     |
#        and the width will be returned.                                  |
#        Case          Returned list                                      |
#        density=F     weight  , width                                    |
#        dimension>=3  weight  , width                                    |
#        dimension=1   x(grid) , y(lsmde), weight  , width                |
#        dimension=2   x(xgrid), y(ygrid), z(lsmde), weight, width        |
#                                                                         |
# EXAMPLE                                                                 |
#       dyn.load("lswtunx.o")       # Dynamic loading of object code    |
#       contour(lsm(scott.density,h=c(1,1)))  # 2-dim lsmde contour       |
#       plot(lsm(buffalo.dat,h=45),type="l")                              |
#-------------------------------------------------------------------------+
lsm<-function(x,ng=50,h=rep(0.96*n^(-1/(id+4)),id), from, to,
              density=T,eps=1e-12)
{
  if(is.vector(x)){ n <- length(x); id <- 1        }
  if(is.matrix(x)){ n <- nrow(x)  ; id <- ncol(x)  }
  if(n<3)stop("Too small data")
  wtlist <- lswt(x,h=h,eps=eps)
  wt     <- wtlist$weight
  obf    <- wtlist$obf
  if(density==F)return(list(weight=wt,width=h,obf=obf))
  else{
    if(id>=3){
      warning("Dimension is too high to calculate")
```

```
      warning("Use brush() to examine the density")
      return(list(weight=wt,width=h,obf=obf))
    }
    if(id==2){
      if(missing(from))from <- apply(x,2,min)-0.75*h[1]
      if(missing(to))  to   <- apply(x,2,max)+0.75*h[2]
      xg  <- seq(from[1],to[1],length=ng)
      yg  <- seq(from[2],to[2],length=ng)
      nz  <- seq(1,n)[wt>eps]
      npw <- length(nz)
      px  <- x[nz,]
      pwt <- wt[nz]
      ax  <- outer(xg, px[,1],FUN="dxmake",h=h[1])
      ay  <- outer(yg, px[,2],FUN="dxmake",h=h[2])
      fh  <-
          matrix(dnorm(ax),ng,npw) %*% (pwt*t(matrix(dnorm(ay),ng,npw)))
      fh  <- fh/(h[1]*h[2])
      return(list(x=xg,y=yg,z=fh,weight=wt,width=h,obf=obf))
    }
    if(id==1){
      if(missing(from))from <- min(x)-0.75*h
      if(missing(to))  to   <- max(x)+0.75*h
      xg  <- seq(from[1],to[1],length=ng)
      nz  <- seq(1,n)[wt>eps]
      npw <- length(nz)
      px  <- x[nz]
      pwt <- wt[nz]
      ax  <- outer(xg, px,FUN="dxmake",h=h)
      fh  <- (matrix(dnorm(ax),ng,npw) %*% pwt)[,1]
      fh  <- fh/h
      return(list(x=xg,y=fh,weight=wt,width=h,obf=obf))
    }
  }
}

#-------------------------------------------------------------------------+
# lswt   Calculating the optimized weights for the LSMDE                  |
#        using Fortran routine QLD.F                                      |
#                                                                         |
# ARGUMENT                                                                |
#     x: vector/matrix of observations from f(x).                         |
#     h: bandwidth. Default is the reference value for the standard       |
#        normal density 0.96*n^(-1/id+4). (see Silverman(1986, p87)),     |
#        where id is the dimension of x.                                  |
#   eps: epsilon for QL0001 for machine precision. Default is 1e-12.      |
#                                                                         |
# VALUE                                                                   |
#        List of two component - wt, h, suitable as input for lsmde       |
#        calculation.                                                     |
```

```
#    wt: vector of optimized wts according to data.                    |
#     h: bandwidth selected by user or default.                        |
#                                                                      |
# EXAMPLE                                                              |
#       dyn.load("lswtunx.o")       # Dynamic loading of object code   |
#       wts <- lswt(scott.density, h=c(1.0,1.0), eps=1e-7)$wt          |
#       print(round(wts,5))         # Print optimized weights          |
#----------------------------------------------------------------------+
lswt <- function(x,h=rep(0.96*n^(-1/(id+4)),id),eps=1e-12)
{
# STEP 1 : Allocate storage/Setup Constraints
  if(is.vector(x)){ n <- length(x); id <- 1      }
  if(is.matrix(x)){ n <- nrow(x)  ; id <- ncol(x)}
  if(n<3)stop("Too small data ")
  if(id>=5)warning("Too high dimension")
  mmax <- n+2
  m    <- n+1
  a    <- matrix(0,mmax,n)
  c    <- matrix(0,n,n)
  lwar <- 3*n*n/2+10*n+2*mmax+1
  mnn  <- m+n+n
  storage.mode(x) <- "double"
  storage.mode(h) <- "double"
  storage.mode(c) <- "double"
  storage.mode(a) <- "double"
  wt   <- vector("numeric",n)
  z    <-.Fortran("weight",
        n               ,#(N    )
        as.integer(id)  ,#(ID   )
        as.double(x)    ,#(X    )
        as.double(h)    ,#(H    )
        double(id)      ,#(H2   )
        as.integer(mmax),#(MMAX)
        as.double(a)    ,#(A    )
        double(mmax)    ,#(B    )
        as.double(c)    ,#(C    )
        double(n)       ,#(D    )
        as.integer(mnn) ,#(MNN  )
        double(mnn)     ,#(U    )
        as.integer(lwar),#(LWRK)
        double(lwar)    ,#(WRK  )
        double(n)       ,#(XL   )
        double(n)       ,#(XU   )
        double(id)      ,#(XI   )
        double(id)      ,#(XJ   )
        integer(n)      ,#(IWRK)
        wt=as.double(wt),#(WT   )
        obf=double(1)   ,#(OBF  )
        ierr=integer(1) ,#(IERR)
```

```
        as.double(eps)    #(EPS )
  )
  cat("IERRCODE:",z$ierr,"\n")
  switch(z$ierr+1,
    print("Success (No Error Detected)"),
    warning("Too many iterations"),
    warning("Insufficient accuracy"),
    stop("Failure (See QPL error code)")
  )
  z1<-list(weight=z$wt,h=h,obf=z$obf)
  z1
}
#-------------------------------------------------------------------------+
# dlsm1    Calculating the ULSMDE/ELSMDE                                  |
#                                                                         |
# ARGUMENT                                                                |
#     x: vector of observations from f(x).                                |
#     g: grid points for discretizations.                                 |
#     h: bandwidth. Default is the reference value for standard normal    |
#        density 0.96*n^(-1/5).                                           |
#    ng: points where density is estimated (Default=50).                  |
#  from: starting point where density is estimated.                       |
#    to: end point where density is estimated.                            |
#  flag: If flage==2, ULSMDE is calculated.                               |
#        Else ELSMDE will be returned.                                    |
#                                                                         |
# VALUE                                                                   |
#        List of x, y, wt, C.                                             |
#                                                                         |
# EXAMPLE                                                                 |
#       plot(dlsm1(mydata, flag=1)) # Plot of ULSMDE for mydata           |
#-------------------------------------------------------------------------+
dlsm1 <- function(x,g=seq(minx-h,maxx+h,length=n),
         h=1.06*n^(-0.2)*sqrt(var(x)),ng=50, from=g[1], to=g[m], flag=1)
{
  fh   <- NULL
  x    <- x[!is.na(x)]
  minx <- min(x)
  maxx <- max(x)
  n    <- length(x)
  m    <- length(g)
  hh   <- sqrt(2)*h
  k0   <- dnorm(0,sd=hh)
  C    <- diag(k0,m,m)
  for(k in (1:(m-1))){
    kij <- dnorm(g[1]-g[k+1],sd=hh)
    for(i in (1:(m-k))){
      C[i,i+k] <- kij
      C[i+k,i] <- kij
```

```
    }
  }
  C  <- round(C,4)
  d  <- apply(apply(x %o% rep(1,m) - rep(1,n) %o%
g,2,dnorm,sd=h),2,mean)
  xg    <- seq(from,to,length=ng)
  if(flag == 2){
    a  <- solve(C,d)
    for(i in (1:ng)){
      fh[i] <- sum(a*dnorm(xg[i],mean=g,sd=h))
    }
    print(h)
    return(list(x=xg, y=fh, weight=a, width=h, tt=C))
  }
  else {
    Cinv  <- solve(C)
    tmp1  <- as.vector(Cinv %*% d)
    tmp2  <- as.vector(Cinv %*% rep(1,m))
    const <- (sum(tmp1)-1)/sum(tmp2)
    a     <- as.vector(Cinv%*%d - const*tmp2)
    for(i in (1:ng)){
      fh[i] <- sum(a*dnorm(xg[i],mean=g,sd=h))
    }
    print(h)
    return(list(x=xg, y=fh, weight=a, width=h,tt=C))
  }
}

#-------------------------------------------------------------------------+
# exp.mixture2.mom  Estimate parameters of a mixture of 2 exponentials   |
#                   via Method of Moments (Rider (1961,AMS))              |
#                                                                         |
# ARGUMENT                                                                |
#     x: vector of observations from f(x)=a mixture of 2 exponentials,    |
#        f(x) = p r1 exp(-r1*x) + (1-p) r2 exp(-r2*x)                     |
#        where p, r1(>0), r2(>0) are all unknown, x>0                     |
#ignore: option for bypassing the conditon check on scale parameters.     |
#                                                                         |
# VALUE                                                                   |
#        List of three component - r1, r2, p. when condition met          |
#    b1: inverse of rate parameter of the first component.                |
#    b2: inverse of rate parameter of the second componet.                |
#     p: mixing proportion.                                               |
#                                                                         |
#        The sequence {t0/t1, t1/t2, t2/t3} should be either increasing   |
#        or decreasing to assure r1, r2 to be positive                    |
#                                                                         |
# NOTE                                                                    |
#       Rider(1961) gives correct quadratic equation for the problem      |
```

```
#       Example in Everitt and Hand (p.62, 1981) provides wrong answer.
#       Quadratic equation in Titterington at. al. (1985) is incorrect.
#
# EXAMPLE
#       exp.mix.mom(exp.mix1)
#
#-----------------------------------------------------------------------+
exp.mixture2.mom <-function(x,ignore=F)
{
  n  <- length(x)
  t0 <- 1
  t1 <- mean(x)
  t2 <- (sum(x^2)/n)/2
  t3 <- (sum(x^3)/n)/6

  c1 <- t0/t1
  c2 <- t1/t2
  c3 <- t2/t3
  is.monotone
      <- ( (c1<= c2) && (c2 <= c3) ) || ( (c3 <= c2) && (c2 <= c1) )
  if(is.monotone || ignore){
      a1 <- (t1*t2-t0*t3)/(t1^2-t0*t2)
      a2 <- (t1*t3-t2^2)/(t1^2-t0*t2)
      b1 <- (a1 - sqrt(a1^2+4*a2))/2
      b2 <- (a1 + sqrt(a1^2+4*a2))/2
      p  <- (t1-b2)/(b1-b2)
      return(list(b1=b1,b2=b2,p=p))
  }
  else stop("At least one of scale parameters is negative\n")
}
#-----------------------------------------------------------------------+
# exp.mixture.em    Estimate parameters of a mixture of k exponentials
#                   via EM algorithm
#
# ARGUMENT
#     x: vector of observations from f(x)=a mixture of 2 exponentials,
#        f(x) = p r1 exp(-r1*x) + (1-p) r2 exp(-r2*x)
#        where p, r1(>0), r2(>0) are all unknown, x>0
#    r0: initial vector of rate parameter.
#    p0: initial vector of mixing proportion.
#detail: If TRUE, then each iteration result will be printed with flag
#        (flag=0, if the iteration is correct)
# VALUE
#       List of three component - r1, r2, p. when condition met
#  rate: vector of rate parameter.
#  prop: mixing proportion.
#
# EXAMPLE
#       exp.mixture.em(x,r0=c(1,2),p0=c(0.5,0.5),detail=T))
```

```
#-----------------------------------------------------------------------+
exp.mixture.em <- function(x,r0=1,p0=1/k,maxiter=50,tol=1e-6,detail=F)
{
  n      <- length(x)
  k      <- length(r0)
  phi0   <- -1/r0
  iter   <- 0
  if(detail){
    xmean <- mean(x)
    cat("exp.mixture.em() : initial parameters rate0 = ",round(r0,6),
        ", prob0 = ",round(p0,6),fill=T)
  }
  repeat{
   iter    <- iter+1
   tmp     <- rep(1,k) %o% x
   tmp     <- apply(tmp,2,dexp,rate=r0)*p0
   w       <- scale(tmp,center=F,scale=apply(tmp,2,sum))
   p1      <- apply(w,1,mean)
   phi1    <- as.vector(-(w%*%x)/(n*p1))
   r1      <- -1/phi1
   maxdiff <- max(max(abs(r0-r1)),max(abs(p0-p1)))
   if(maxdiff<tol)break
   if(iter>maxiter)stop("Does not converge in maximum iterations")
   r0    <- r1
   p0    <- p1
   if(detail){
     cat("Iteration ",iter)
     cat(" : rate = ",round(r0,6)," prob = ",round(p0,6))
     iszero <- round(sum(p0*phi1)+xmean,4)
     cat(" flag = ",iszero,fill=T)
   }
  }
  print("Convergence met in exp.mixture.em()")
  return(list(rate=as.vector(r1), prop=p1))
}
```

From QLD.F

```
      SUBROUTINE QL0001(M,ME,MMAX,N,NMAX,MNN,C,D,A,B,XL,XU,
     1           X,U,IOUT,IFAIL,IPRINT,WAR,LWAR,IWAR,LIWAR)
C
C                    !!!! NOTICE !!!!
C
C 1. The routines contained in this file are due to Prof. K.Schittkowski
C    of the University of Bayreuth, Germany (modification of routines
C    due to Prof. MJD Powell at the University of Cambridge).  They can
C    be freely distributed.
C
C 2. A minor modification was performed at the University of Maryland.
```

```
c     It is marked in the code by "c umd".
c
c                                        A.L. Tits and J.L. Zhou
c                                        University of Maryland
C
C
C
C               SOLUTION OF QUADRATIC PROGRAMMING PROBLEMS
C
C
C
C     QL0001 SOLVES THE QUADRATIC PROGRAMMING PROBLEM
C
C     MINIMIZE          .5*X'*C*X + D'*X
C     SUBJECT TO        A(J)*X  +  B(J)    =  0  ,  J=1,...,ME
C                       A(J)*X  +  B(J)   >=  0  ,  J=ME+1,...,M
C                       XL  <=  X  <=  XU
C
C     HERE C MUST BE AN N BY N SYMMETRIC AND POSITIVE MATRIX, D AN N-
C     DIMENSIONAL VECTOR, A AN M BY N MATRIX AND B AN M-DIMENSIONAL
C     VECTOR. THE ABOVE SITUATION IS INDICATED BY IWAR(1)=1.
C     ALTERNATIVELY, I.E. IF IWAR(1)=0,THE OBJECTIVE FUNCTION MATRIX CAN
C     ALSO BE PROVIDED IN FACTORIZED FORM.IN THIS CASE, C IS AN UPPER
C     TRIANGULAR MATRIX.
```

References

Ash, R. B. (1972). *Real Analysis and Probability.* Academic Press Inc. London.

Bartels, R. H., Golub, G. H., and Saunders, M. A. (1970). "Numerical Techniques in Mathematical Programming" in *Nonlinear Programming* edited by Rosen, J. B., Morgasarin, O. L., and Ritter, K. (1970), Academic Press, New York.

Becker, R. A., Chambers, J. M., and Wilks, A. R. (1988). *The New S Language.* Wadsworth & Brooks/Cole, Pacific Grove, CA.

Bowman, A. W. (1984). "An Alternative Method of Cross-Validation for the Smoothing of Density Estimates" *Biometrika* **71** 353-360.

Bowman, A. W., Hall, P., and Titterington, D. M. (1984). "Cross-validation in Nonparametric Estimation of Probabilities and Probability Densities." *Biometrika* **71** 341-351.

Bowman, A. W. (1985) "A Comparative Study of Some Kernel-Based Nonparametric Density Estimators." *J. Statist. Comput. Simul.* **21** 313-327.

Chambers, J. M. and Hastie, T. J. (1991). *Statistical Models in S.* Wadsworth & Brooks/Cole, Pacific Grove, CA.

Draper, N. R. and Smith, H. (1981). *Applied Regression Analysis.* 2nd Ed. John Wiley & Sons, New York.

Dhamadhikari, S. W. and Joag-dev, K. (1988). *Unimodality, convexity, and Applications* Academic Press, Boston.

Devroye, L. and Györfi, L. (1985). *Nonparametric Density Estimation: The L_1 View.* John Wiley, New York.

Everitt, B. S. and Hand, D. J. (1981). *Finite Mixture Distributions.* Chapman and Hall, New York

Fan, J. and Marron, J. S. (1994). "Fast Implementations of Nonparametric Curve Estimators" *Journal of Computational and Graphical Statistics* **3** 35-56.

Feller, W. (1966). *An Introduction to Probability Theory and Its Applications.* Vol. 2, John Wiley, New York.

Fletcher, R. (1987). *Practical Methods of Optimization.* John Wiley & Sons, New York.

Gasser, T., Engel, J., and Seifert. (1993). "Nonparametric Function Estimation" in *Handbook of Statistics,* Vol. 9 : *Computational Statistics* edited by Rao, C. R.

(1993), Elsevier Science Publishers, Amsterdam, North-Holland.

Geman, S. (1981). "Sieves for Nonparametric Estimation of Densities and Regressions" Reports in Pattern Analysis No 99., Division of Applied Mathematics, Brown University, Providence, Rhode Island.

Geman, S. and Hwang, C. R. (1982). "Nonparametric Maximum Likelihood Estimation by the method of sieves" *Ann. Statist.* **41**, 1344-1346.

Gill, P. E., Murray, W., and Wright, M. H. (1981). *Practical Optimization.* Academic Press Inc. London.

Goldfarb, D., and A. Idnani (1983), A numerically stable dual method for solving strictly convex quadratic programs, *Mathematical Programming*, **27**, 1-33.

Golub, G. H. and Van Loan, C. F. (1983). *Matrix Computations*. The Johns Hopkins Univ. Press, Baltimore, MD.

Good, I. J. and Gaskins, R. A. (1980). "Density Estimation and Bump-Hunting by the Penalized Likelihood Method Examplified by Scattering and Meteorite Data" *Journal of the American Statistical Association*, **75**, 42-73.

Good, I. J. and Holtzman, G. I. (1989). "Diagmosis of Heart Attack from two Enzyme Measurements by means of bivariate probability density estimation: Statistical Details" *Journal of Computational and Graphical Statistics* **32** 68-76.

Hager, W. W., Hosrt, R., and Pardalos, P. M. (1993). "Mathematical Programming – A Computational Perspective" in *Handbook of Statistics*, Vol. 9 : *Computational Statistics* edited by Rao, C. R. (1993), Elsevier Science Publishers, Amsterdam North-Holland.

Härdle, W. (1990). *Smoothing Techniques with Implementation in S.* Springer-Verlag, Springer-Verlag, New York.

Härdle, W. and Scott, D. W. (1992). "Smoothing by Weighted Averaging of Shifted Points" Computational Statistics **7**, 97-128.

IMSL (1991). *IMSL/MATH User's Guide*, IMSL Inc., Texas

Karlin, S. (1968). *Total Positivity*. Stanford University Press, Stanford, CA.

Lawson, C. L. and Hanson, R. J. (1974). *Solving Least Squares Problems*. Prentice-Hall, Engelwood Cliffs, NJ.

Lukacs, E. (1960). *Characteristic Functions.* Hafner Publishing Company, New York.

McLachlan, G. J. (1992). *Discriminant Analysis and Statistical Pattern Recognition.* John Wiley & Sons, New York.

Minotte, M. C. and Scott, D. W. (1993). "The Mode Tree: A Tool for Visualization of Nonparametric Density Features" *Journal of Computational and Graphical Statistics* **2** 51-68.

Powell, M. J. D. (1983). *ZQPCVX a FORTRAN subroutine for convex quadratic programming,* DAMTP Report NA17, Cambridge, England.

Powell, M. J. D. (1985). On the quadratic programming algorithm for Goldfarb and Idnani, *Mathematical Programming Study,* **25**, 46-61.

Redner, R. A. and Walker, H. F. (1984). "Mixture Densities, Maximum Likelihood and the EM algorithm" *SIAM Review* **26** 195-239.

Rudemo, M. (1982). "Empirical Choice of Histograms and Kernel Density Estimators" *Scandinavian Journal of Statistics* **9** 65-78.

Schittkowski, K. (1980). *Nonlinear Programming Codes*, Springer-Verlag, New York.

Scott, D. W. (1976). "Nonparametric Probability Density Estimation by Optimization Theoretic Techniques" Technical Report No. 476-131-1, Rice University, Houston, Texas.

Scott, D. W., Gotto, A. M., Cole, J. S., and Gorry, G. A. (1978). "Plasma Lipids as Collateral Risk Factors in Coronary Artery Disease: A Study of 371 Males with Chest Pain" *Journal of Chronic Diseases* **31** 337-345.

Scott, D. W. and Terrell, G. R. (1987) "Biased and Unbiased Cross-Validation in Density Estimation" *J. Amer. Statist. Assoc.* **82**, 1131-1146

Scott, D. W. (1992). *Multivariate Density Estimation: Theory, Practice, and Visualization.* John Wiley & Sons, New York.

Silverman, B. W. (1981). "Using Kernel Density Estimates to Investigate Multimodality" *J. Roy. Statist. Soc. B* **43** 93-99.

Silverman, B. W. (1986). *Density Estimation for Statistics and Data Analysis.* Chapman and Hall, London.

Tapia, R. A. and Thompson, J. R. (1978). *Nonparametric Probability Density Estimation.* Johns Hopkins University Press, Baltimore.

Terrell, G. R. and Scott, D. W. (1980). "On Improving Convergence Rates for Non-negative Kernel Density Estimators" *Ann. Statist.* **8**, 1160-1163

Terrell, G. R. and Scott, D. W. (1992). "Variable Kernel Density Estimation" *Ann. Statist.* **20**, 1236-1265

Thisted, R. A. (1988). *Elements of Statistical Computing*. Chapman and Hall, New York.

Titterington, D. M., Smith, A. F. M., and Makov, U. E. (1985). *Statistical Analysis of Finite Mixture Distributions*. John Wiley, New York.

Williams, E. J. (1959). *Regression Analysis*. Wiley, New York.

Vita

Donggeon Kim was born in Taegu, Korea, on February 13, 1964. He received his B.A. in Business Administration in 1986 from Yonsei University, Seoul, Korea. He entered Virginia Polytechnic Institute and State University in 1988 and received his M.A. in Statistics in 1991, and his Ph.D. in Statistics in 1995.