# Lec 11: Alternative density estimation methods

Weiping Zhang

November 24, 2020

# Nearest neighbor density estimation

- The basic idea of the nearest neighbor method is to control the degree of smoothing in the density estimate based on the size of a box required to contain a given number of observations.
- The size of this box is controlled using an integer $k$, that is considerably smaller than the sample size, a typical choice would be $k \approx n^{1/2}$.
- For any point $x$ on the line we define the distance between $x$ and the points on the sample by

$$d_i(x) = |x_i - x|$$

The observations ranked by the distances, or "nearest neighbors", $x_{(1)} \leq \cdots \leq x_{(n)}$. so that

$$d_1(x) \leq d_2(x) \leq \cdots \leq d_n(x)$$

- Then we define the $k$th nearest neighbor density estimate by

$$\hat{g}(x) = \frac{k}{2nd_k(x)} = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{2d_k(x)}I(|x - X_i| \leq d_k(x))$$

- Intuitively, if $d_k(x)$ is small this means that there are many observations near $x$; so $g(x)$ must be large, while if $d_k(x)$ is large this means that there are not many observations near $x$; so $g(x)$ must be small
- A motivation for this estimator is that the effective number of observations to estimate $g(x)$ is $k$, which is constant regardless of $x$.
- The nearest neighbor estimator is not smooth: $d_k(x)$ has a discontinuity in its derivative at every point $x_i$.

- Furthermore, although $\hat{g}(x)$ is positive and continuous everywhere it is not in fact a probability density. Outside $[x_{(1)}, x_{(n)}]$, we get $d_k(x) = x_{(k)} - x$ and $d_k(x) = x - x_{(n-k+1)}$ which make the tails of the $\hat{g}(x)$ fall off like $x^{-1}$, that is extremely slowly: the integral of $\hat{g}(x)$ is infinite.
- This can in principle be fixed by using a generalized $k$th nearest neighbor estimate

$$\hat{g}(x) = \frac{1}{n d_k(x)} \sum_{i=1}^{n} K(\frac{x - X_i}{d_k(x)})$$

In fact this is just a kernel estimate evaluated at $x$ with window width $d_k(x)$ (Adaptive KDE)

# Maximum Penalized Likelihood Estimators

- The methods discussed so far are all derived in an ad hoc way from the definition of a density.
- It is interesting to ask whether it is possible to apply standard statistical techniques, like maximum likelihood, to density estimation. The *likelihood* of a curve $g$ as density underlying a set of independent identically distributed observations is given by

$$L(g|X_1, \ldots, X_n) = \prod_{i=1}^{n} g(X_i)$$

- This likelihood has no finite maximum over the class of all densities. To see this, let $\hat{f}_h$ be the naive density estimate with window width $1/2h$, then, for each $i$,

$$\hat{f}_h(X_i) \geq \frac{1}{nh}$$

and so

$$\prod_{i=1}^{n} \hat{f}_h(X_i) \geq n^{-n} h^{-n} \to \infty, \quad as\ h \to 0$$

- Thus the likelihood can be made arbitrarily large by taking densities approaching the sum of delta functions, and it is not possible to use maximum likelihood directly for density estimation without placing restrictions on the class of densities over which the likelihood is to be maximized.

- There are, nevertheless, possible approaches related to maximum likelihood. One method is to incorporate into the likelihood a term which describes the roughness - in some sense - of the curve under consideration. Suppose $R(g)$ is a functional which quantifies the roughness of $g$. One possible choice of such a functional is

$$R(g) = \int (g'')^2$$

- Define the penalized log likelihood by

$$l_\lambda(g) = \sum_{i=1}^{n} log(g(X_i)) - \lambda R(g)$$

where $\lambda$ is a positive smoothing parameter.

- The penalized log likelihood can be seen as a way of quantifying the conflict between smoothness and goodness-of-fit to the data, since the log likelihood term sum $log(g(X_i))$ measures how well $g$ fits the data.
- The probability density function hat $f$ is said to be a maximum penalized likelihood density estimate if it maximizes $l_\lambda(g)$ over the class of all curves $g$ which satisfy $\int g = 1$, $g(x) \geq 0$ for all $x$, and $R(g) < \infty$.
- The parameter $\lambda$ controls the amount of smoothing since it determines the 'rate of exchange' between smoothness and goodness-of-fit; the smaller the value of $\lambda$, the rougher - in terms of $R(\hat{f})$ - will be the corresponding maximum penalized likelihood estimator.
- Estimates obtained by the maximum penalized likelihood method will, by definition, be probability densities.

# Orthogonal series density estimation

- Let $X$ be a random variable with pdf $f \in L_2(R)$, and $\{\varphi_j(x)\}$ be the orthogonal basis of $L_2(R)$, then

$$f(x) = \sum_{j=0}^{\infty} \theta_j \varphi_j(x)$$

  where $\theta_j = \int_R \varphi_j(x) f(x) dx$.
- Note that $\theta_j = E\varphi_j(X)$, then $\theta_j$ can be estimated under $i.i.d$ sample $X_1, \ldots, X_n$ by

$$\hat{\theta}_j = \frac{1}{n} \sum_{i=1}^{n} \varphi_j(X_i)$$

- The commonly used orthogonal basis include Hermite basis, Laguerre basis, Fourier basis, orthogonal polynomials and wavelets etc.
- Which orthogonal series should be used depends on the support of $f$. When $(-\infty, \infty)$ or $(0, \infty)$ are the support, the Hermite and Laguerre series are recommended. If $f$ has a compact support, the Fourier series are commonly choose.
- Assume that the random variable $X$ is supported on $[0, 1]$, we consider the following Fourier basis

$$\{\varphi_0(x) = 1, \varphi_j(x) = \sqrt{2}cos(\pi jx), j = 1, 2, \ldots\}$$

and $\hat{\theta}_j$ is an unbiased estimator of $\theta_j$, that is

$$E\hat{\theta}_j = \int_0^1 \varphi_j(x) f(x) dx = \theta_j$$

- The variance is easily calculated with the help of the elementary trigonometric identity $cos^2(\alpha) = [1 + cos(2\alpha)]/2$ which allows us to write

$$Var(\hat{\theta}_j) = n^{-1}[1 + 2^{-1/2}\theta_{2j} - \theta_j^2] =: n^{-1}d_j$$

  where $d_j$ is called the coefficient of difficulty.

- Fourier coefficients of any square integrable density decrease as $j$ increases due to the Parseval's identity

$$\int_0^1 f^2(x)dx = 1 + \sum_{j=1}^{\infty} \theta_j^2$$

- One of the attractive features of orthogonal series estimation is the simplicity of considering multivariate densities. Furthermore, both continuous and discrete components can be considered.

Any orthogonal series estimator can be written as

$$\hat{f}(x) = \hat{f}(x, \{\hat{w}_j\}) = 1 + \sum_{j=1}^{\infty} \hat{w}_j \hat{\theta}_j \varphi_j(x) \tag{1}$$

where $\hat{w}_j \in [0, 1]$ is a shrinking coefficient.

- **Truncated Estimators** These are estimation procedures mimicking (1) with $\hat{w}_j = I(j \leq J)$. Denote a truncated density estimator as

$$\tilde{f}_J(x) = 1 + \sum_{j=1}^{J} \hat{\theta}_j \varphi_j(x)$$

then using the Parseval identity and the variance expression equation, we have

$$E\int_0^1 [\tilde{f}_J(x) - f(x)]^2 dx = \sum_{j=1}^{J} Var(\hat{\theta}_j) + \sum_{j=J+1}^{\infty} \theta_j^2$$

$$= n^{-1}\sum_{j=1}^{J} d_j + \sum_{j=J+1}^{\infty} \theta_j^2$$

- A cutoff $J+1$ is worse than the cutoff $J$ if $n^{-1}d_{J+1} > \theta_{J+1}^2$.

Of course, the unbiased estimator of $\theta_j^2$ is $\hat{\theta}_j^2 - n^{-1}d_j$.

- Tarter and Kronmal (1976) proposed to choose $J$ as a minimal integer $J$ such that $2n^{-1}d_{J+i} > \hat{\theta}_{J+i}$ for all $i = 1, \ldots, r$.

Specifically, $r = 2$ is recommended.

- Diggle and Hall (1986), Hart (1985) suggests others cutoffs.

- **Threshold Estimators** This is a more complicated and also potentially more rewarding estimation procedure.
  - Neither asymptotic theory nor numerical simulations support $\hat{w}_j = I(\hat{\theta}_j^2 > 2d_j n^{-1})$
  - Two types of thresholding have been proposed. *Hard thresholding* use weights

$$\hat{w}_j = I(|\hat{\theta}_j| > t(j,n)\sqrt{d_j n^{-1}})$$

    with $t(j,n)$ being a specific function; *Soft thresholding* use weights

$$\hat{w}_j = \frac{|\hat{\theta}_j| - t(j,n)\sqrt{d_j n^{-1}}}{|\hat{\theta}_j|} I(|\hat{\theta}_j| > t(j,n)\sqrt{d_j n^{-1}})$$

  - In simulations, these two procedures perform similarly but soft thresholding is simpler for the theoretical analysis. The most popular threshold is $t(j,n) = \sqrt{2}[ln(n)]^{1/2}$.

- **Mimicking of Oracle** This is based on the idea of asking an oracle about optimal shrinking of the empirical Fourier coefficients.
  - Since $min_w E(\theta_j - w_j\hat{\theta}_j)^2 = E(\theta_j - w_j^*\hat{\theta}_j)^2$ where

$$w_j^* = \frac{\theta_j^2}{\theta_j^2 + d_j n^{-1}}$$

  - Then it is natural to use a statistic in place of $w_j^*$, for instance, the unbiased estimate $\hat{\theta}_j^2 - d_j n^{-1}$ of $\theta_j^2$ can be plugged in. Good numerical outcomes have been reported, but it is also possible to show that any estimator based on term-by-term shrinkage is not asymptotically minimax.
  - To overcome this issue, a blockwise shrinkage should be used:

$$min_W \sum_{j \in B} E(\theta_j - W\hat{\theta}_j)^2 = \sum_{j \in B} E(\theta_j - W^*\hat{\theta}_j)^2$$

where

$$W^* = \frac{\sum_{j \in B} \theta_j^2}{\sum_{j \in B} [\theta_j^2 + d_j n^{-1}]}$$

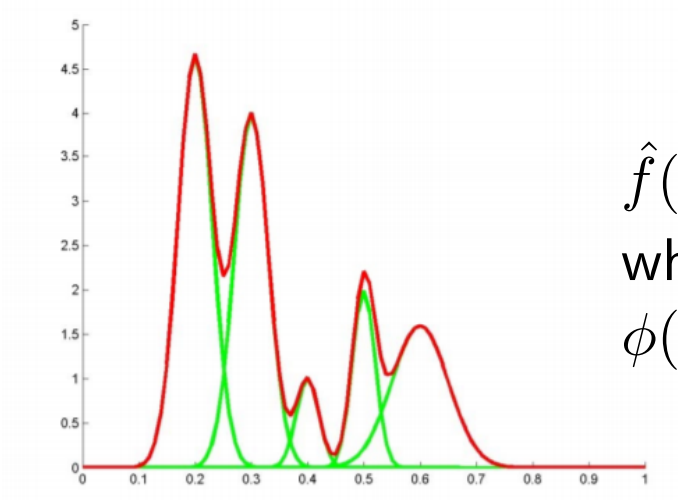
- **Universal Estimator** It combines main underlying ideas of the above introduced estimators. The first step is to calculate a pilot estimate $\tilde{f}(x) = 1 + \sum_{j=1}^{\hat{J}} \hat{w}_j \hat{\theta}_j \varphi_j(x)$ where $\hat{w}_j = \max(0, 1 - d_j/n\hat{\theta}_j^2)$ and $\hat{J}$ minimizes $\sum_{j=1}^{J}[2d_j/n - \hat{\theta}_j^2]$
  - The first one is based on the idea of obtaining a good estimator for spatially inhomogeneous densities that may have several relatively large Fourier coefficients beyond the cutoff $\hat{J}$.

$$\breve{f}(x) = \tilde{f}(x) + \sum_{j=\hat{J}+1}^{c_{JM}\hat{J}} I(\hat{\theta}_j^2 > c_T d_j ln(n)/n)\hat{\theta}_j \varphi_j(x)$$

  - The second improvement is projecting the above modified estimate onto a class of nonnegative densities: $\hat{f} = \max(0, \breve{f}(x) - c)$ where $c$ is such that $\int_0^1 \hat{f}(x)dx = 1$.

# Mixture density estimation

- It looks like we could do better by just picking the right # of Gaussians:




$\hat{f}(x) = \sum_{s=1}^{C} \phi(x; \mu_s, \sigma_s^2)\pi_s$
where $\pi_s \geq 0, \sum_{s=1}^{C} \pi_s = 1$ and
$\phi(x; \mu_s, \sigma_s^2)$ is the density of $N(\mu_s, \sigma_s^2)$.

- This is indeed a good model: density is multimodal because there is a hidden variable $Z$ which determinates the mixture components, the resulting density is a "mixture of Gaussians"

## mixtures vs kernels

- The general mixture density estimate can be written as

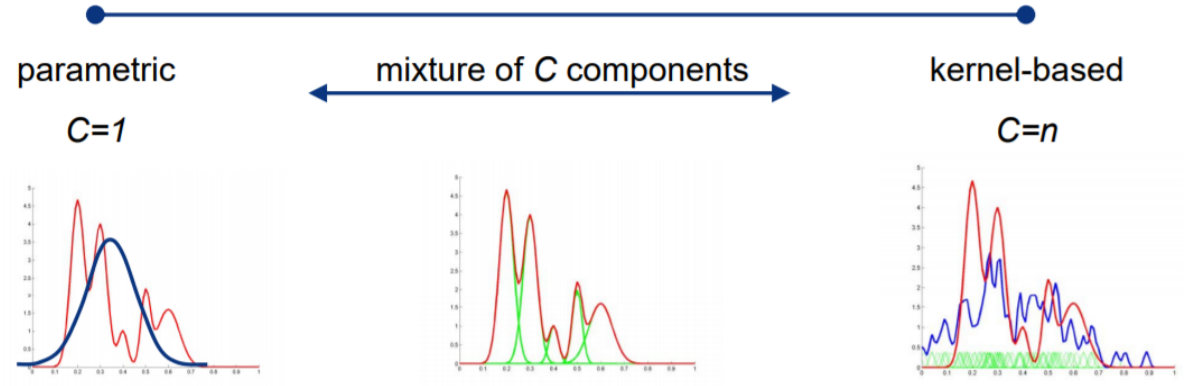
$$\hat{f}(x) = \sum_{s=1}^{C} \pi_s p(x; \theta_s)$$

- This looks a lot like kernel density estimate

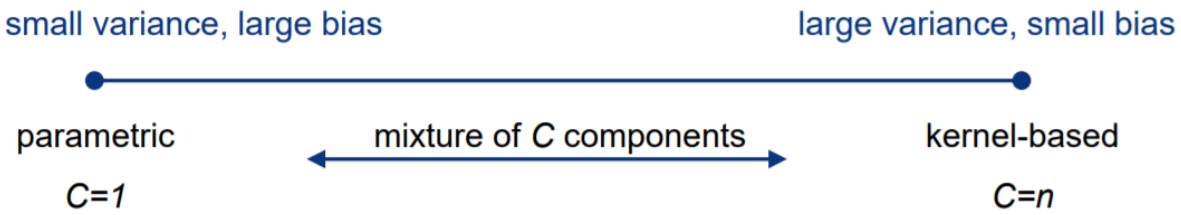
$$\hat{f}_h(x) = \frac{1}{n} \sum_{i=1}^{n} K_h(X_i - x)$$

which is a mixture of $n$ components by uniform weight $1/n$

- mixtures provide a connection between parametric model and KDE:




- with respect to parametric estimates
  more degrees of freedom (parameters) $\Rightarrow$ less bias
- with respect to kernel estimates
  much smaller # of components $\Rightarrow$ less parameters, less variance



# mixture disadvantages

- main disadvantage is learning complexity
- nonparametric estimates: simple: store the samples (NN); place a kernel on top of each point (kernel-based)
- parametric estimates: small amount of work: if ML equations have closed-form; substantial amount of work: otherwise (numerical solution)
- mixtures:
  - there is usually no closed-form solution
  - always need to resort to numerical procedures
- standard tool is the expectation maximization (EM) algorithm
- to see this let's consider gaussian mixture of $C$ components example: the parameters $\Psi = \{(\pi_1, \mu_1, \Sigma_1), \ldots, (\pi_C, \mu_C, \Sigma_C)\}$

- The complete log-likelihood function becomes

$$l(\Psi) = \sum_{k=1}^{n}\sum_{i=1}^{C} log f(y_k, z_i) = \sum_{k=1}^{n}\sum_{i=1}^{C} z_i[log\pi_i + ln\phi(y_k; \mu_i, \Sigma_i)]$$

$$\propto \sum_{k=1}^{n}\sum_{i=1}^{C} z_i[log\pi_i - \frac{1}{2}ln|\Sigma_i| - \frac{1}{2}tr(\Sigma_i^{-1}(y_k - \mu_i)(y_k - \mu_i)')]$$

- E step:

$$Q(\Psi|\Psi^t) = E[l(\Psi)|\mathbf{y}, \Psi^t]$$

$$= \sum_{k=1}^{n}\sum_{i=1}^{C} E[z_i|\mathbf{y}, \Psi^t][log\pi_i - \frac{1}{2}ln|\Sigma_i|$$

$$- \frac{1}{2}tr(\Sigma_i^{-1}(y_k - \mu_i)(y_k - \mu_i)')] + const$$

where $E[z_i|y_k, \Psi^t] = \frac{\pi_i^t \phi(y_k; \mu_i^t, \Sigma_i^t)}{\sum_{l=1}^{C} \pi_l^t \phi(y_k; \mu_l^t, \Sigma_l^t)} =: \hat{z}_{ik}$

- M step:

$$\hat{\pi}_i^{t+1} = \frac{1}{n}\sum_{k=1}^{n} \hat{z}_{ik}$$

$$\hat{\mu}_i^{t+1} = \frac{\sum_{k=1}^{n} \hat{z}_{ik} y_i}{\sum_{k=1}^{n} \hat{z}_{ik}}$$

$$\hat{\sum}_i^{t+1} = \frac{\sum_{k=1}^{n} \hat{z}_{ik}(y_k - \mu_i^{t+1})(y_k - \mu_i^{t+1})'}{\sum_{k=1}^{n} \hat{z}_{ik}}$$

- The EM algorithm alternates between the E step and the M step until convergence.

# FFT-Based Algorithms for Kernel Density Estimation

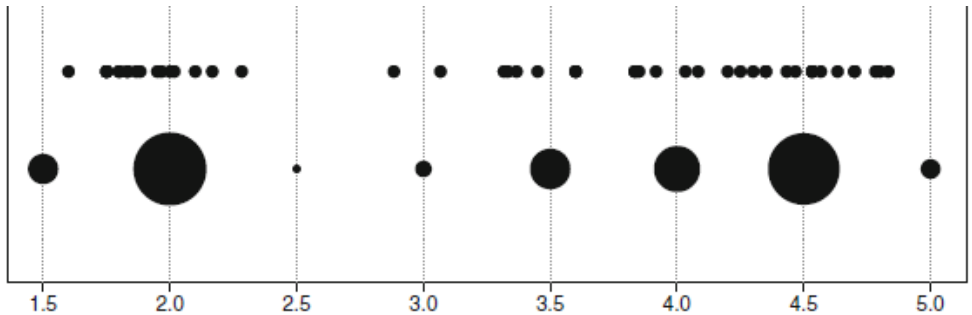
- Analyzing individual formulas for KDE and bandwidth selection, it is not difficult to notice that $O(n^2)$ computational complexity is a common case.

$$\hat{f}_h(x) = \frac{1}{n} \sum_{i=1}^{n} K_h(x - X_i),$$

- Consequently, for large datasets, naive computations are a very bad decision.
- Fast Fourier Transform (FFT), which allows a huge computational speedups without a loss of accuracy, can serve as a tool for density estimation with huge datasets

# FFT-Based Algorithms for Kernel Density Estimation

- The natural step could be to make use of binning, that is for every sample point $X_i$ to be replaced by a pair of two values: the grid point $g_i$ and the grid count $c_i$



input data (upper circles) / output binned data (lower circles)

- The following example is based on a sample bivariate dataset Unicef, available from the ks R package. Each data point, shown as small filled circles, is replaced by a grid of equally spaced grid points (gray filled circles) of sizes $5 \times 8$, $10 \times 10$ and $20 \times 20$.

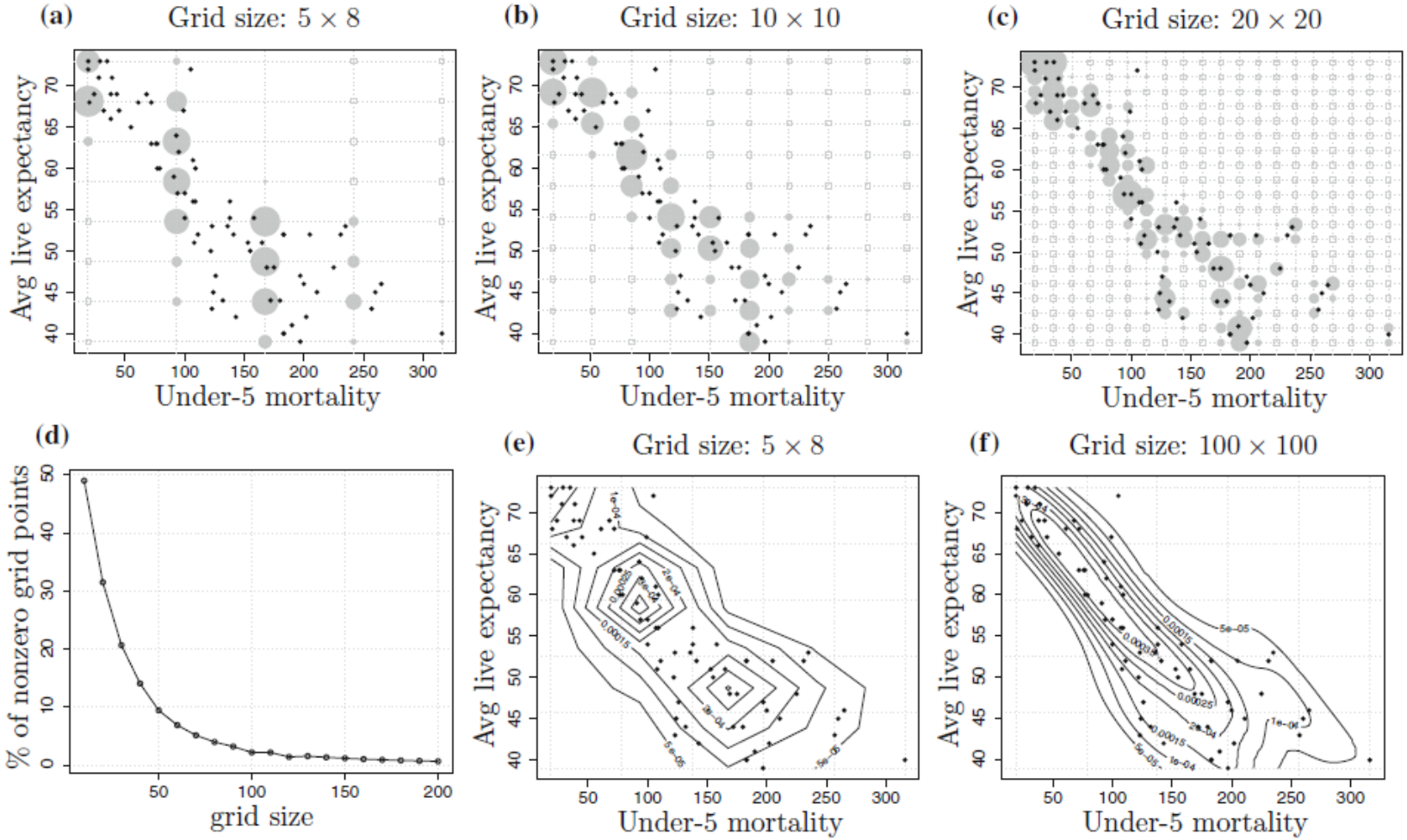



**Fig. 5.5** An illustrative example of data binning. A sample 2D dataset *Unicef* is binned

- In practical applications, it is more desirable to compute KDE for equally spaced grid points $g_j$ where $j = 1, \ldots, M$, For the univariate case, $M \approx 400 - 500$ seems to be absolutely sufficient in terms of most of the applications. The KDE can be obviously rewritten as

$$\hat{f}_j \equiv \hat{f}\left(g_j, h\right) = \frac{1}{n} \sum_{i=1}^{n} K_h\left(g_j - X_i\right), \quad j = 1, \ldots, M$$

- Since every sample point $X_i \to (g_i, c_i)$, the ordinary KDE can obviously rewritten as

$$\hat{f}_j(x) = \frac{1}{n} \sum_{l=1}^{M} K_h(g_j - g_l) c_l, \quad l = 1, \ldots, M \qquad (2)$$

- Since the kernel is symmetric and the grid points are equally spaced, the number of kernel evaluation is $O(M(M+1)/2)$. While the number of multiplications $K_h(\cdot)c_k$ is still $O(M^2)$. To reduce this value to $O(Mlog_2M)$, the FFT-based technique can be used.
- To use the FFT for a fast computation of (2), this equation must be rewritten again as

$$\hat{f}_j = \frac{1}{n}\sum_{l=1}^{M} K_h(g_j - g_l)c_l = \sum_{l=1}^{M} k_{j-l}c_l \qquad (3)$$

where

$$k_{j-l} = \frac{1}{n}K_h(\delta(j-l)), \delta = \frac{b-a}{M-1}$$

with $a = g_1, b = g_M$ and $\delta$ is the grid width.

- The second summation in (3) has not yet had the form of the 'pure' convolution. The goal is to get a convolution-like equation, which can be solved very fast using the FFT algorithm.
- In order to represent it as convolution, we should first observe that

$$c_l = 0 \quad for\ l \notin \{1, \ldots, M\}$$

and

$$K(-x) = K(x)$$

- In that case, the summation can be safely extended to $-M$, that is

$$\hat{f}_j = \sum_{l=-M}^{M} k_{j-l} c_l = \mathbf{c} * \mathbf{k}$$

where $*$ is the convolution operator.

- Observe that the factors for $l = -M$ and $l = M$ are always zeroed out, then (3) can be rewritten as

$$\hat{f}_j = \sum_{l=-(M-1)}^{M-1} c_{j-l} k_l \tag{4}$$

  where $k_l = \frac{1}{n} K_h(\delta l)$.

- In (4) the vector $\hat{f}_j$ has the character of a discrete convolution of two vectors $\mathbf{c} = (c_1, \ldots, c_M)$ and $\mathbf{k} = (k_{-(M-1)}, \ldots, k_{-1}, k_0, \ldots, k_{M-1})$ and can be calculated very effectively using the well-known FFT algorithm.
- The discrete convolution theorem places certain requirements on the form of vectors $\mathbf{c}$ and $\mathbf{k}$. Since the two lengths here are not the same, the special procedure known as *padding the signal with zeros* (often abbreviated as *zero-padding*) needs to be employed.(For details, see reading materials)
- The FFT-based approach was implemented by R *ks* package.

# Selecting the bandwidth using bootstrap

- It focuses on replacing the MSE by MSE*, a bootstrapped version of MSE, which can be minimized directly
- Some authors resample from a subsample of the data $X_1, ..., X_n$; others replace from a pilot density based on the data, more precisely, from

$$\tilde{f}_h^b(x) = \frac{1}{nb_n} \sum_{i=1}^{n} L(\frac{x - X_i}{b_n})$$

where $L$ is another kernel and $b_n$ is a pilot bandwidth.

- Since the bandwidth choice reduces to estimating $s$ in $h = n^{-1/5}s$, Ziegler (2006) introduces

$$f^*_{n,s}(x) = \frac{1}{n^{4/5}s}\sum_{i=1}^{n} K(\frac{x - X^*_i}{n^{-1/5}s})$$

and obtain $MSE^*_{n,s}(x) = E^*((f^*_{n,s}(x) - \tilde{f}^b_h(x))^2)$. The proposed bandwidth is

$$h_n = n^{-1/5} \arg\min_s MSE^*_{n,s}$$

- Applications of the bootstrap idea can be found in many different areas of estimation, see Delaigle and Gijbels (2004), Loh and Jang (2010) for example.