L7: Kernel density estimation

Non-parametric density estimation

Histograms

Parzen windows

Smooth kernels

Product kernel density estimation

The naïve Bayes classifier

Density Functions

Suppose we have some variable $X \sim f(x)$ where $f(x)$ is the probability density function (pdf) of X.

Note that we have two requirements on $f(x)$:

- $f(x) \geq 0$ for all $x \in \mathcal{X}$, where $\mathcal{X}$ is the domain of X
- $\int_{\mathcal{X}} f(x) dx = 1$

Example: normal distribution pdf has the form

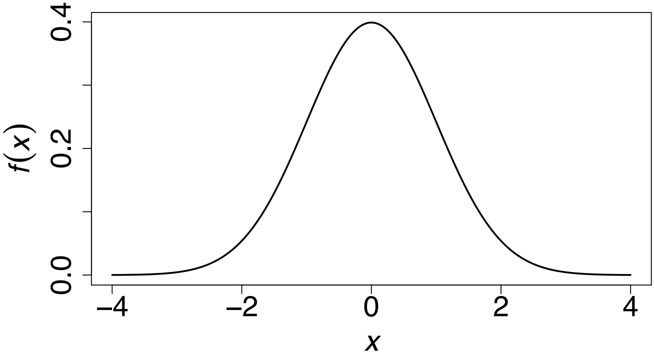
$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

which is well-defined for all $x, \mu \in \mathbb{R}$ and $\sigma \in \mathbb{R}^+$.

Standard Normal Distribution

If $X \sim \mathrm{N}(0,1)$, then X follows a standard normal distribution:

$$f(x) = \frac{1}{\sqrt{2\pi}} e^{-x^2/2} \quad (1)$$



Non-parametric density estimation

Sometimes we assume either

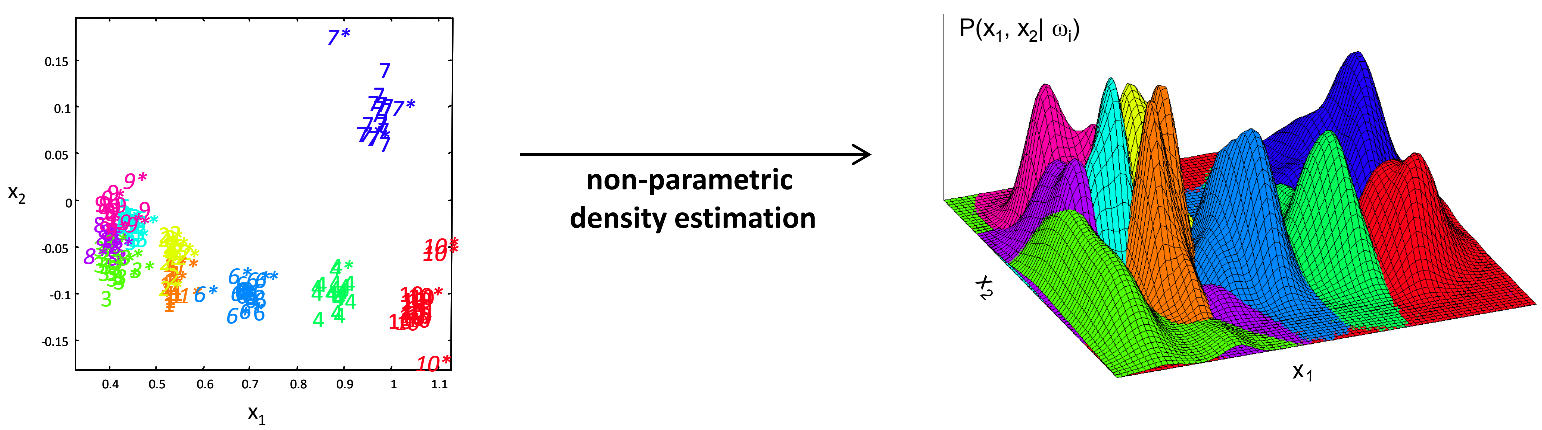
– The likelihoods $p(x|\omega_i)$ were known (LRT), or

– At least their parametric form was known (parameter estimation)

But in a lot of cases, we do not afford such luxuries

– Instead, they attempt to estimate the density directly from the data without assuming a particular form for the underlying distribution

– Sounds challenging? You bet!



The histogram

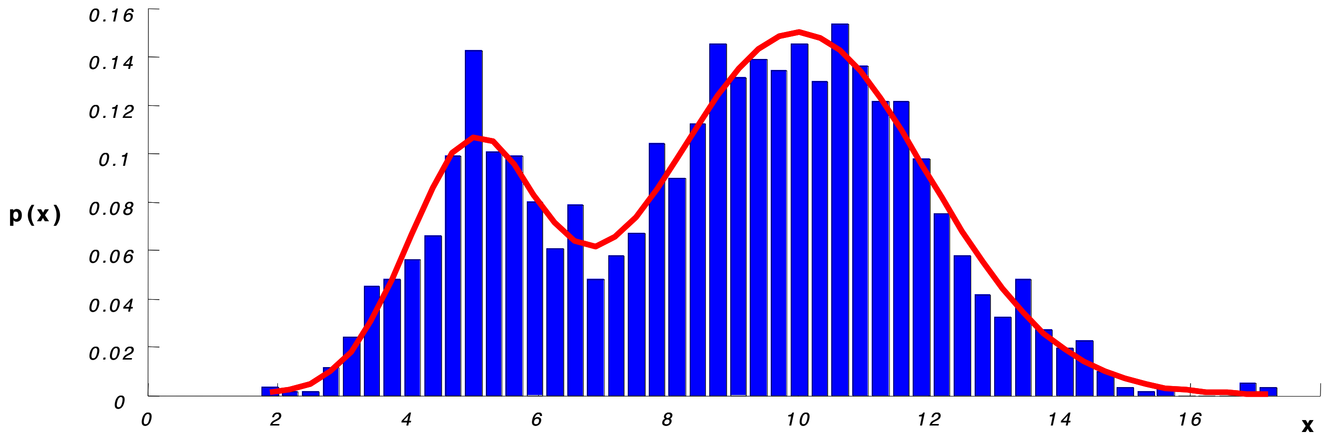
The simplest form of non-parametric DE is the histogram

– Divide the sample space into a number of bins and approximate the density at the center of each bin by the fraction of points in the training data that fall into the corresponding bin

$$p_H(x) = \frac{1}{N}\frac{[\#\ of\ x^{(k}\ in\ same\ bin\ as\ x]}{[width\ of\ bin]}$$

N =total number of points

– The histogram requires two "parameters" to be defined: <u>bin width</u> and <u>starting position</u> of the first bin



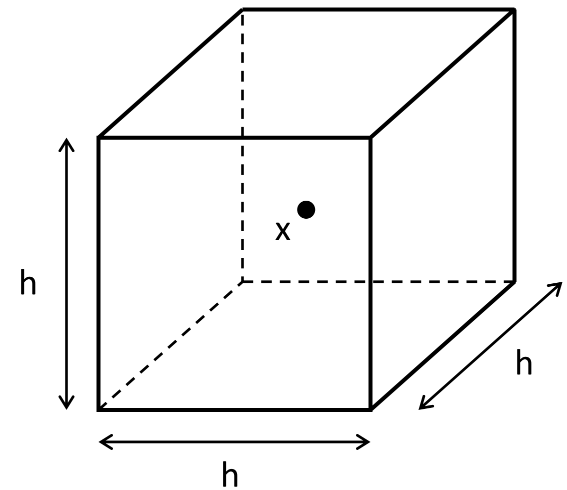
The histogram is a very simple form of density estimation, but has several drawbacks

- The density estimate depends on the starting position of the bins
  - For multivariate data, the density estimate is also affected by the orientation of the bins
- The discontinuities of the estimate are not due to the underlying density; they are only an artifact of the chosen bin locations
  - These discontinuities make it very difficult (to the naïve analyst) to grasp the structure of the data
- A much more serious problem is the curse of dimensionality, since the number of bins grows exponentially with the number of dimensions
  - In high dimensions we would require a very large number of examples or else most of the bins would be empty
- These issues make the histogram unsuitable for most practical applications except for quick visualizations in one or two dimensions
- Therefore, we will not spend more time looking at the histogram

Parzen windows

Problem formulation

– Assume that the region $\Re$ that encloses the k examples is a hypercube with sides of length h centered at x
  - Then its volume is given by $V = h^D$, where D is the number of dimensions



– To find the number of examples that fall within this region we define a kernel function $K(u)$

$$K(u) = \begin{cases} 1 & |u_j| < 1/2 \quad \forall j = 1 \ldots D \\ 0 & otherwise \end{cases}$$

  - This kernel, which corresponds to a unit hypercube centered at the origin, is known as a Parzen window or the naïve estimator
  - The quantity $K((x - x^{(n})/h)$ is then equal to unity if $x^{(n}$ is inside a hypercube of side h centered on x, and zero otherwise

[Bishop, 1995]

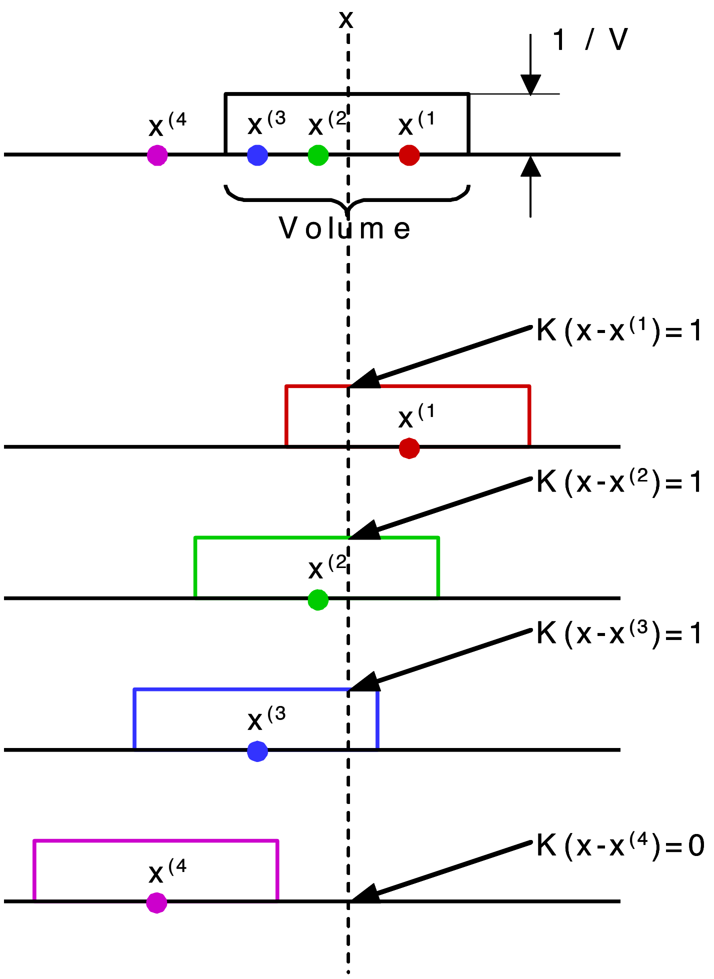
– The total number of points inside the hypercube is then

$$k = \sum_{n=1}^{N} K\left(\frac{x - x^{(n}}{h}\right)$$

Substituting back into the expression for the density estimate

$$p_{KDE}(x) = \frac{1}{Nh^D} \sum_{n=1}^{N} K\left(\frac{x - x^{(n}}{h}\right)$$

– Notice how the Parzen window estimate resembles the histogram, with the exception that the bin locations are determined by the data



Smooth kernels

The Parzen window has several drawbacks

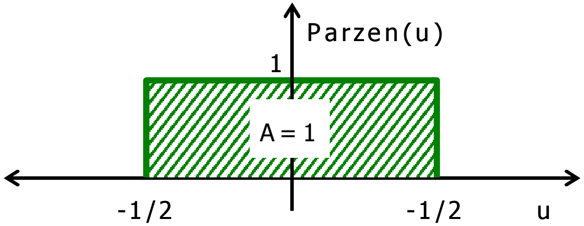
- It yields density estimates that have discontinuities
- It weights equally all points x_i, regardless of their distance to the estimation point x

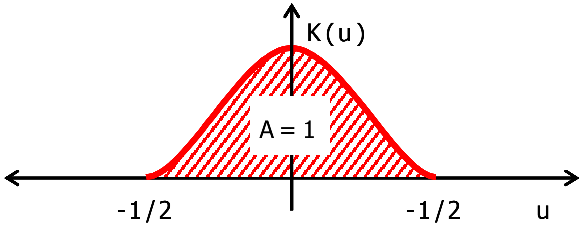
For these reasons, the Parzen window is commonly replaced with a smooth kernel function $K(u)$

$$\int_{R^D} K(x)dx = 1$$

- Usually, but not always, $K(u)$ will be a radially symmetric and unimodal pdf, such as the Gaussian $K(x) = (2\pi)^{-D/2} e^{-\frac{1}{2}x^T x}$
- Which leads to the density estimate

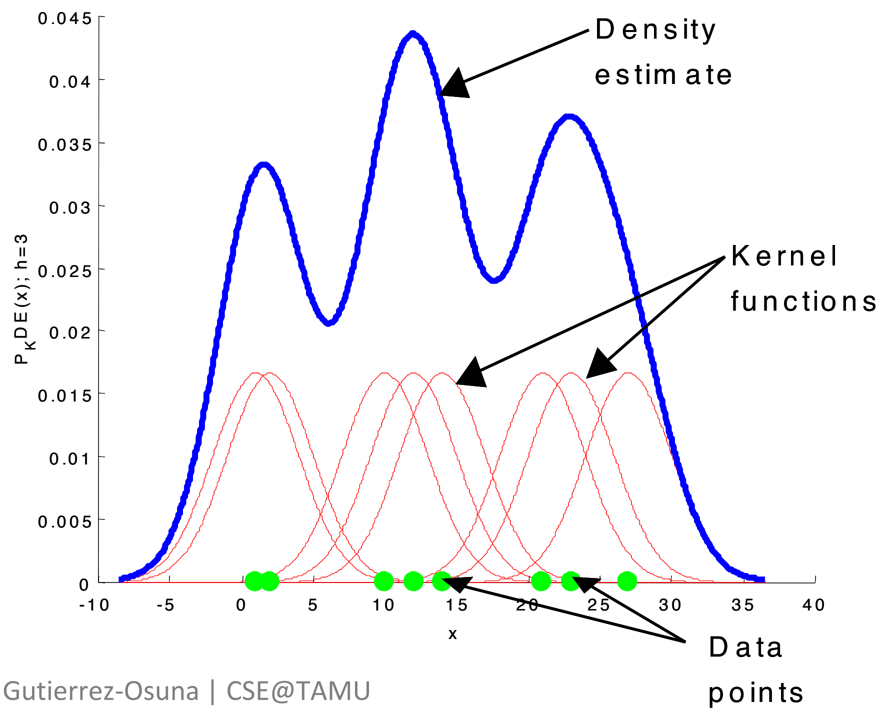
$$p_{KDE}(x) = \frac{1}{Nh^D} \sum_{n=1}^{N} K\left(\frac{x-x^{(k}}{h}\right)$$





Interpretation

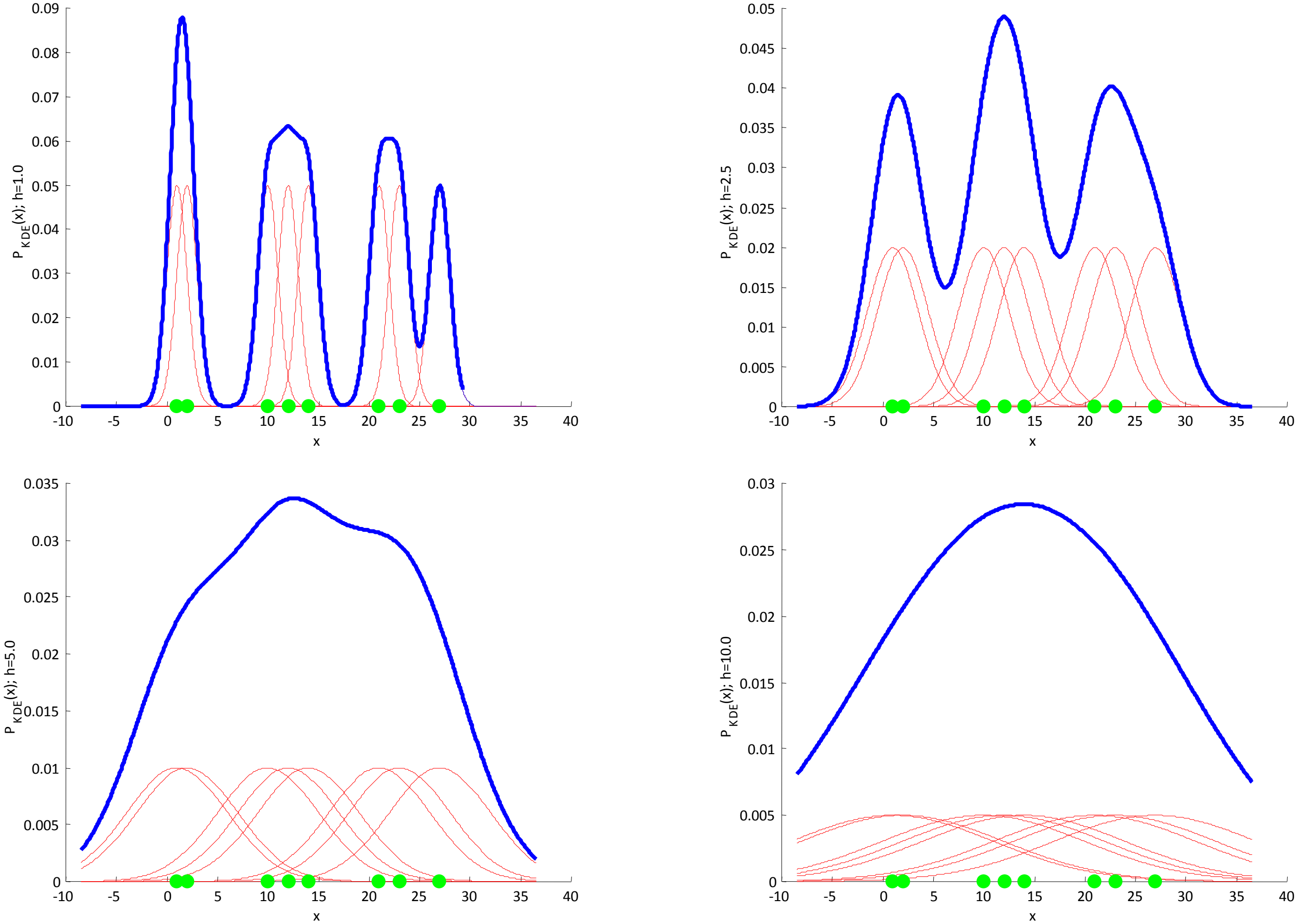
- Just as the Parzen window estimate can be seen as a sum of boxes centered at the data, the smooth kernel estimate is a sum of "bumps"
- The kernel function determines the shape of the bumps
- The parameter h, also called the <u>smoothing parameter</u> or <u>bandwidth</u>, determines their width



Bandwidth selection

The problem of choosing h is crucial in density estimation

- A large h will over-smooth the DE and mask the structure of the data
- A small h will yield a DE that is spiky and very hard to interpret



Multivariate density estimation

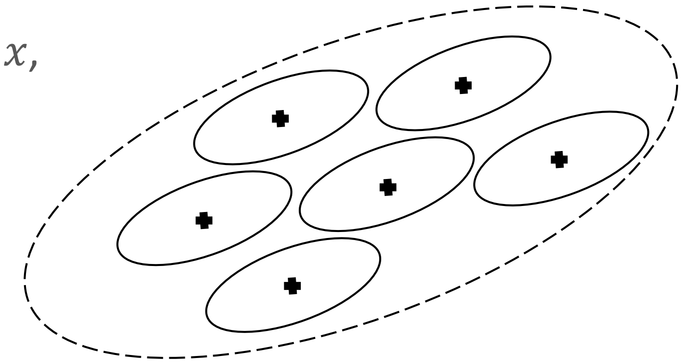
For the multivariate case, the KDE is

$$p_{KDE}(x) = \frac{1}{Nh^D} \sum_{n=1}^{N} K\left(\frac{x - x^{(n}}{h}\right)$$

– Notice that the bandwidth h is the same for all the axes, so this density estimate will weight all the axis equally
– If one or several of the features has larger spread than the others, we should use a vector of smoothing parameters or even a full covariance matrix, which complicates the procedure

There are two basic alternatives to solve the scaling problem without having to use a more general KDE

– Pre-scaling each axis (normalize to unit variance, for instance)
– Pre-whitening the data (linearly transform so $\Sigma = I$), estimate the density, and then transform back [Fukunaga]
  - The whitening transform is $y = \Lambda^{-1/2} M^T x$, where Λ and M are the eigenvalue and eigenvector matrices of Σ
  - Fukunaga's method is equivalent to using a hyper-ellipsoidal kernel



Product kernels

A good alternative for multivariate KDE is the product kernel

$$p_{PKDE}(x) = \frac{1}{N}\sum_{i=1}^{N} K\left(x, x^{(n}, h_1, \dots h_D\right)$$

$$where\ \ K\left(x, x^{(n}, h_1, \dots h_D\right) = \frac{1}{h_1 \dots h_D}\prod_{d=1}^{D} K_d\left(\frac{x_d - x_d^{(n}}{h_d}\right)$$

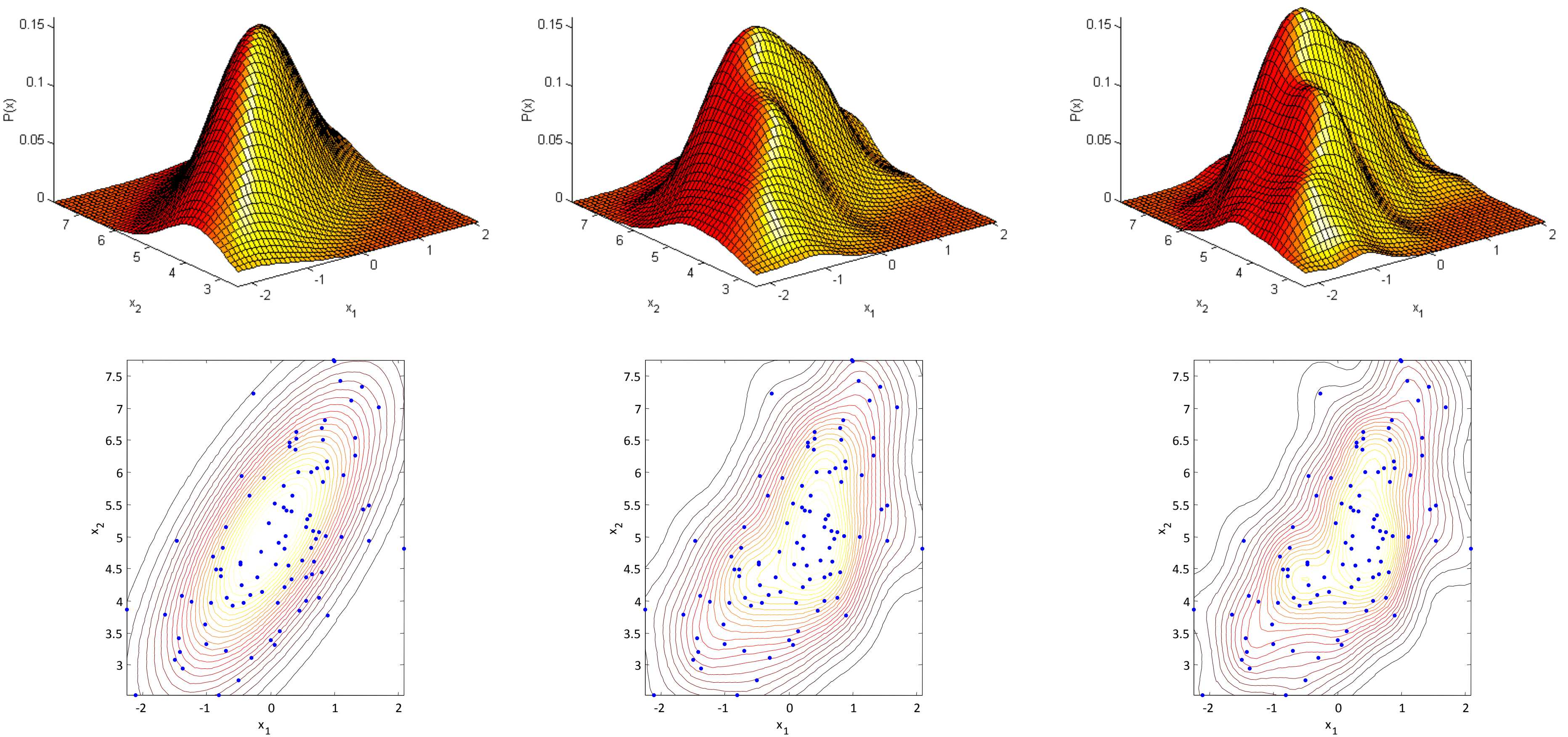
- The product kernel consists of the product of one-dimensional kernels
  - Typically the same kernel function is used in each dimension ($K_d(x) = K(x)$), and only the bandwidths are allowed to differ
  - Bandwidth selection can then be performed with any of the methods presented for univariate density estimation
- Note that although $K\left(x, x^{(n}, h_1, \dots h_D\right)$ uses kernel independence, this does not imply we assume the features are independent
  - If we assumed feature independence, the DE would have the expression

$$p_{FEAT-IND}(x) = \prod_{d=1}^{D} \frac{1}{Nh^D}\sum_{i=1}^{N} K_d\left(\frac{x_d - x_d^{(n}}{h_d}\right)$$

  - Notice how the order of the summation and product are reversed compared to the product kernel

Example I

- This example shows the product KDE of a bivariate <u>unimodal</u> Gaussian
  - 100 data points were drawn from the distribution
  - The figures show the true density (left) and the estimates using $h = 1.06\sigma N^{-1/5}$ (middle) and $h = 0.9AN^{-1/5}$ (right)



Example II

– This example shows the product KDE of a bivariate <u>bimodal</u> Gaussian
  - 100 data points were drawn from the distribution
  - The figures show the true density (left) and the estimates using $h = 1.06\sigma N^{-1/5}$ (middle) and $h = 0.9AN^{-1/5}$ (right)