EECS 553: Machine Learning (ECE) **University of Michigan**

Kernel Density Estimation

Winter 2023 **Clayton Scott**

1 Density Estimation

Density estimation is an unsupervised learning problem where we are given a random sample

$$\boldsymbol{X}_1, \ldots, \boldsymbol{X}_n \sim f$$

where f is an unknown pdf, and the goal is to estimate f. See Figure 1.

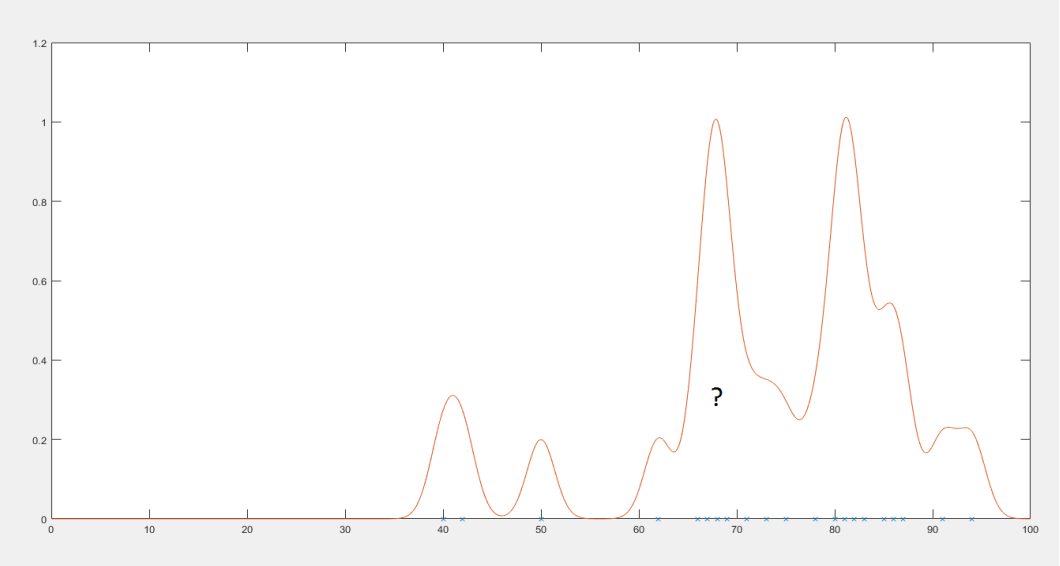

Figure 1: An unknown density that must be estimated from realizations.

Before examining this problem, let's first see why it is important.

1. **Classification:** From the formula for the Bayes classifier, a "plug-in" classifier has the form

$$\boldsymbol{x} \longrightarrow \arg\max_k \widehat{\pi}_k \widehat{g}_k(\boldsymbol{x})$$

where $\widehat{g}_k$ is an estimate of the class-conditional density. Thus, methods of density estimation give rise to plug-in classifiers. The KDE is a nonparametric density estimator and therefore can work effectively in settings where parametric methods like LDA are not suitable.

2. **Clustering:** Clusters can be defined by the modes (local maxima) of the density. In particular, given a point $\boldsymbol{x}$, one can apply gradient ascent to a density estimate until a mode of the density is reached. All $\boldsymbol{x}$ reaching the same mode form a cluster. This is known as *mode-based clustering*, and is commonly implemented using the *mean shift algorithm*.

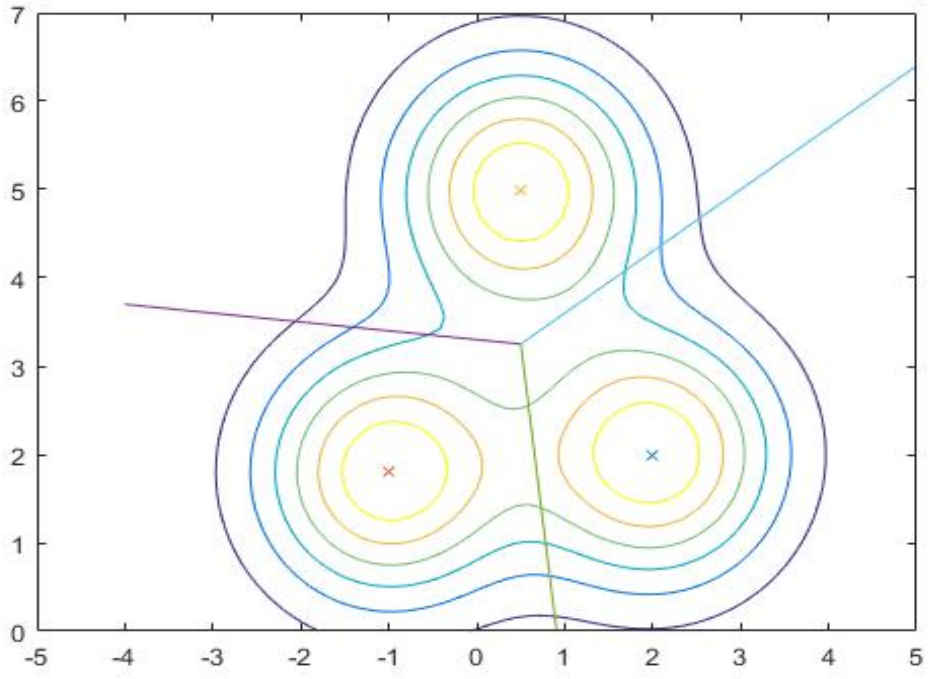

Figure 2: Contours of a density on $\mathbb{R}^2$. The lines indicate the boundaries of clusters according to density-based clustering, but they are not quite drawn in the right places.

3. **Anomaly Detection:** Given $\boldsymbol{X}_1, \ldots, \boldsymbol{X}_n \sim f$, we can form an estimate $\widehat{f}$ of f, and perform the comparison

$$\widehat{f}(\boldsymbol{x}) > \gamma,$$

for some threshold $\gamma > 0$, to decide whether a future observation comes from the same distribution or not. Smaller values of γ mean that a point must be more outlying to be declared anomalous.

2 Kernel Density Estimation

Let $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$ denote n realizations of a random variable $\boldsymbol{X}$ whose distribution is f. A *kernel density estimate* has the form

$$\widehat{f}(\boldsymbol{x}) := \frac{1}{n} \sum_{i=1}^{n} k_\sigma(\boldsymbol{x} - \boldsymbol{x}_i)$$

where $k_\sigma(\boldsymbol{y})$ is called a *kernel*, and $\sigma > 0$ is a parameter called the *bandwidth*.

This notion of kernel is different from others in machine learning, such as symmetric, positive definite kernels and local weighting kernels (as in locally linear regression). Here the kernel k_σ has the form

$$k_\sigma(\boldsymbol{y}) = \sigma^{-d} k\left(\frac{\boldsymbol{y}}{\sigma}\right)$$

where k is usually chosen to satisfy the following properties.

- $\int k(\boldsymbol{y})d\boldsymbol{y} = 1$
- $k(\boldsymbol{y}) \geq 0 \quad \forall \boldsymbol{y} \in \mathbb{R}^d$
- $k(\boldsymbol{y}) = \psi(\|\boldsymbol{y}\|)$ for some $\psi : [0, \infty) \to \mathbb{R}$

A k satisfying the third property is called a *radial kernel*. For example, Gaussian kernel has

$$k(\boldsymbol{y}) = (2\pi)^{-\frac{d}{2}} e^{-\frac{1}{2}\|\boldsymbol{y}\|^2}$$

and the uniform kernel has

$$k(\boldsymbol{y}) = \frac{1}{C}\mathbf{1}_{\{\|\boldsymbol{y}\| \leq 1\}}$$

where C is the volume of the unit sphere in $\mathbb{R}^d$. These and other examples are depicted in Figure 3.

The KDE is sometimes called the Parzen window. It was originally presented by Rosenblatt (1956) and Parzen (1962), although the estimator was actually first proposed in an unpublished technical report by Evelyn Fix and J. L Hodges (1951). For more background see Silverman & Jones, (1989). E. Fix and J.L. Hodges (1951): An Important Contribution to Nonparametric Discriminant Analysis and Density Estimation: Commentary on Fix and Hodges (1951).

The KDE is clearly nonparametric and bears some similarity to nearest neighbor methods. Like nearest neighbor methods, KDEs require no optimization as part of training, but they do require $O(n)$ time to evaluate the density of a test instance. Methods for approximate nearest neighbor seach can in fact be used to speed up the evaluation of KDEs. Furthermore, there are nearest-neighbor methods for density estimation.

So why does it work? The KDE can be viewed as a superposition of shifted kernel function. The more $\boldsymbol{x}_i$ are in a given region of space, the more these shifted kernel accumulate, the larger the estimated density. Figure 4 shows a KDE of midterm exam scores for a past version of EECS 545 (60 points max).

3 Model Selection

The bandwidth σ is a scale parameter that can drastically affect the KDE. See Figure 5.

To perform model selection, we'd like to choose σ to optimize some performance measure. One possible performance measure is the integrated squared error, or L^2 distance,

$$\begin{aligned}
\text{ISE}(\sigma) &= \int (\widehat{f}_\sigma(\boldsymbol{x}) - f(\boldsymbol{x}))^2 d\boldsymbol{x} \\
&= \int \widehat{f}_\sigma(\boldsymbol{x})^2 d\boldsymbol{x} - 2\int \widehat{f}_\sigma(\boldsymbol{x}) f(\boldsymbol{x}) d\boldsymbol{x} + \int f(\boldsymbol{x})^2 d\boldsymbol{x}
\end{aligned}$$

Let's look at these three terms. The last term is independent of σ, so we can ignore it. The first term can be computed explicitly for many kernels. For example, if k_σ is a Gaussian kernel, then

$$\begin{aligned}
\int \widehat{f}_\sigma(\boldsymbol{x})^2 d\boldsymbol{x} &= \frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n}\int k_\sigma(\boldsymbol{x} - \boldsymbol{x}_i) k_\sigma(\boldsymbol{x} - \boldsymbol{x}_j) d\boldsymbol{x} \\
&= \frac{1}{n^2}\sum_{i}\sum_{j} k_{\sqrt{2}\sigma}(\boldsymbol{x}_i - \boldsymbol{x}_j)
\end{aligned}$$

since convolving Gaussian densities amounts to adding independent Gaussian random variables.

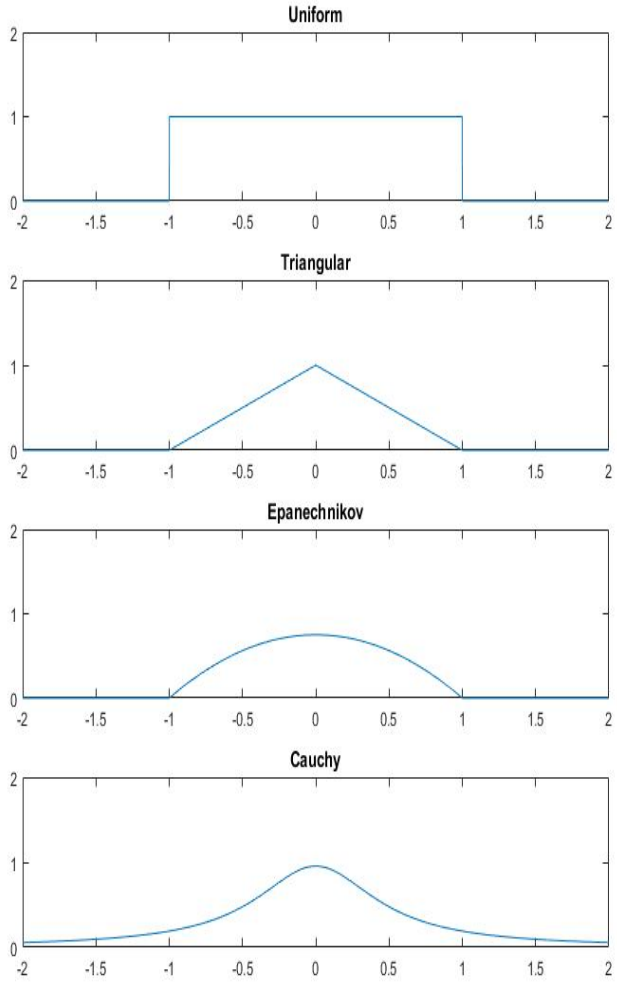

Figure 3: The function ψ defining various radial kernels.

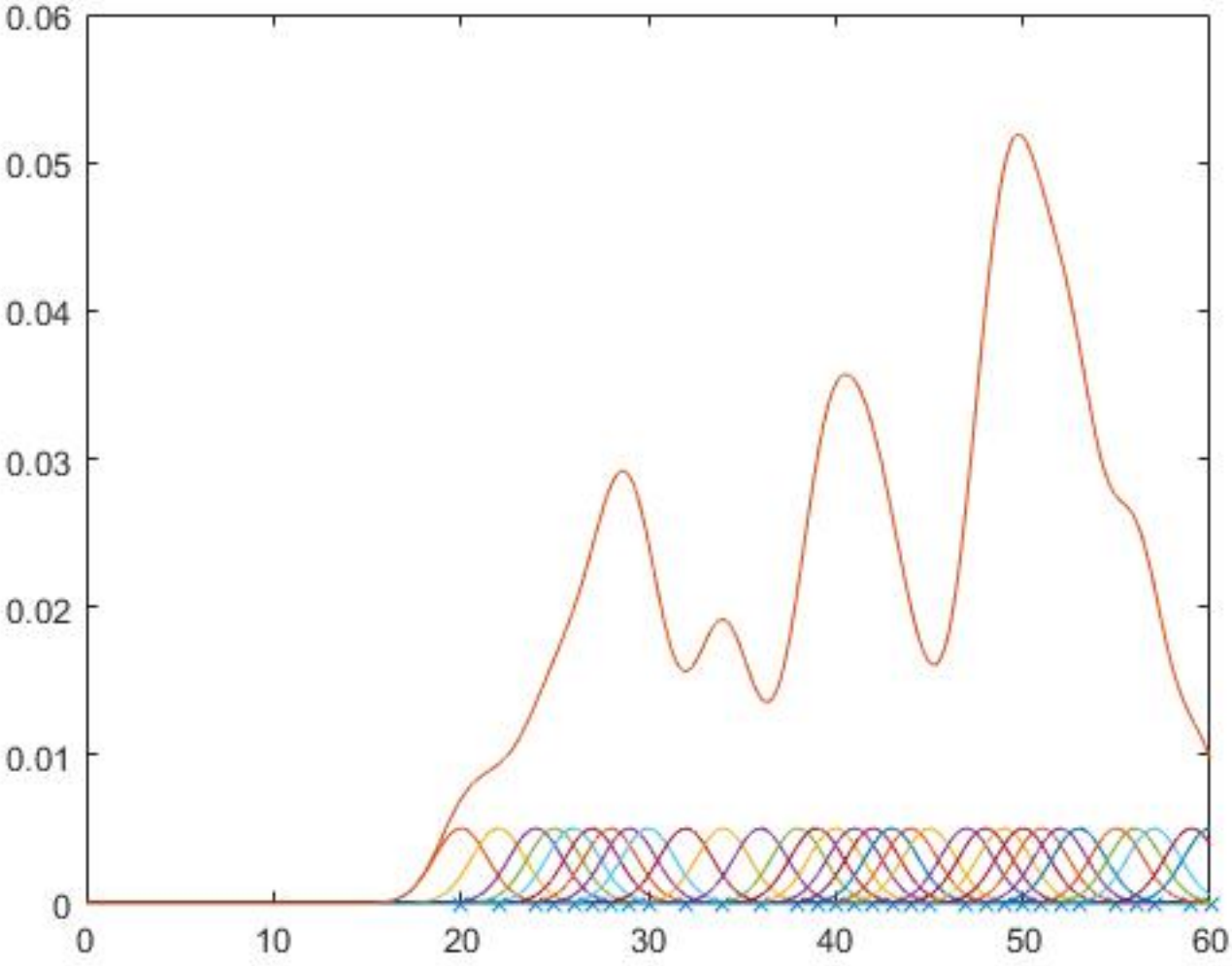

Figure 4: A one dimensional KDE.

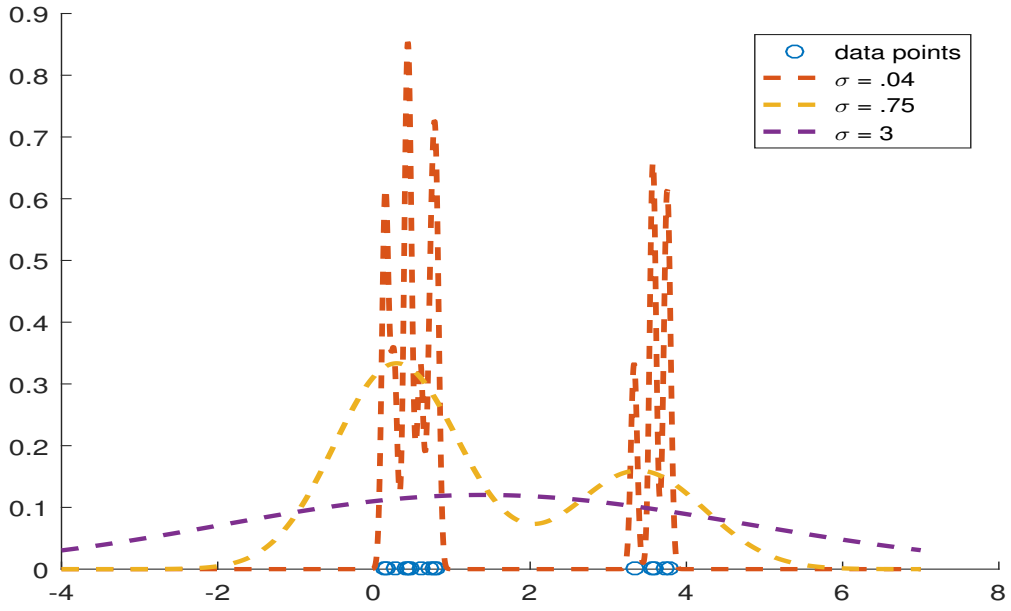

Figure 5: The affect of σ on the KDE.

As for the second term,

$$\int \widehat{f}_\sigma(\boldsymbol{x}) f(\boldsymbol{x}) d\boldsymbol{x} = \mathbb{E}_{\boldsymbol{X}\sim f}[\widehat{f}_\sigma(\boldsymbol{X})]$$

The idea is to estimate expectation using the training data. A simple training error estimate

$$\frac{1}{n}\sum \widehat{f}_\sigma(\boldsymbol{x}_i)$$

would lead to overfitting ($\sigma \to 0$). Instead, it is common to use a leave-one-out estimation

$$\frac{1}{n}\sum_{i=1}^{n} \widehat{f}_\sigma^{-i}(\boldsymbol{x}_i)$$

where

$$\widehat{f}_\sigma^{-i}(\boldsymbol{x}) = \frac{1}{n-1}\sum_{j\neq i} k_\sigma(\boldsymbol{x}-\boldsymbol{x}_j)$$

Putting it all together, this suggests selecting σ (in the case of the Gaussian kernel) by

$$\widehat{\sigma} = \arg\min_\sigma \sum_{i,j=1}^{n} k_{\sqrt{2}\sigma}(\boldsymbol{x}_i, \boldsymbol{x}_j) - \frac{2}{n(n-1)}\sum_{i=1}^{n}\sum_{j\neq i} k_\sigma(\boldsymbol{x}_i, \boldsymbol{x}_j),$$

where the minimization is typically performed over a grid of σ values. The resulting procedure is called LS-LOOCV for least squares leave-one-out cross-validation.