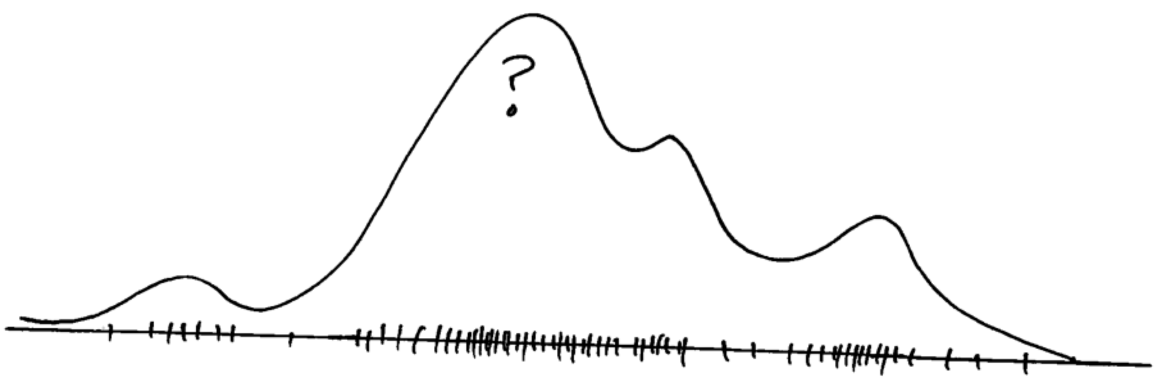
KERNEL DENSITY ESTIMATION

Density Estimation

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

$$X_1, \dots, X_n \sim f$$

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



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

1. Classification

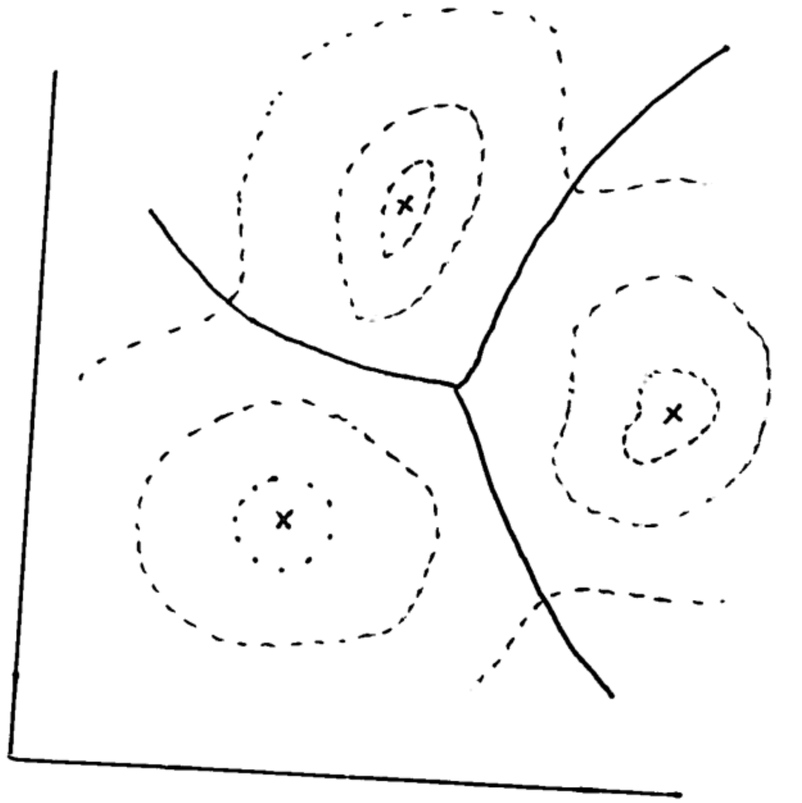
From the formula for the Bayes classifier, a "plug-in" classifier has the form

$$x \longmapsto \arg\max_k \hat{\pi}_k \hat{g}_k(x)$$

where $\hat{g}_k$ is an estimate of the class-conditional density.

2. Clustering

Clusters can be defined by the modes of the density. Given a point x, climb the density until you reach a mode. All x reaching the same mode form a cluster. This is known as mode-based clustering, and is commonly implemented using the mean shift algorithm.



3. Novelty Detection

Given $X_1, \ldots, X_n \sim f$, we can form an estimate $\hat{f}$ of f, and use the detector

$$\hat{f}(x) \gtrless \gamma$$

to decide whether a future observation comes from the same distribution or not.

Kernel Density Estimation

A kernel density estimate has the form

A kernel density estimate has the form

$$\hat{f}(x) := \frac{1}{n} \sum_{i=1}^{n} k_\sigma(x - X_i)$$

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

The kernel k_σ has the form

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

where k is usually chosen to satisfy the following properties.

1. $\int k(y)\, dy = 1$
2. $k(y) \geq 0 \quad \forall y \in \mathbb{R}^d$
3. $k(y) = \psi(\|y\|)$ for some $\psi : [0, \infty) \to \mathbb{R}$. "radial" kernel

Examples

1. Gaussian kernel

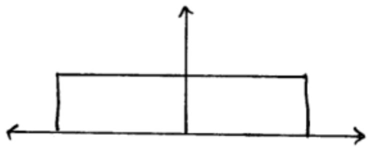
$$k(y) = (2\pi)^{-\frac{d}{2}} \exp\left\{-\frac{1}{2}\|y\|^2\right\}$$

2. Uniform kernel

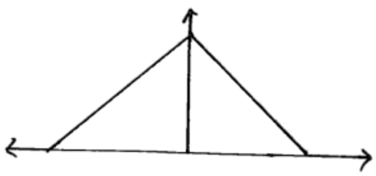
$$k(y) = \frac{1}{C} \cdot \mathbb{1}_{\{\|y\| \leq 1\}}$$

where C is the volume of the unit sphere in $\mathbb{R}^d$.

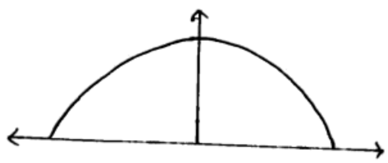
More examples in 1-d:



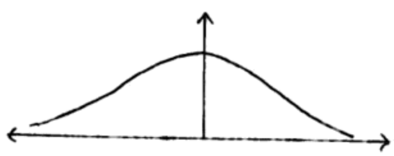
uniform



triangular



Epanichnikov (parabolic)



Cauchy

Remarks

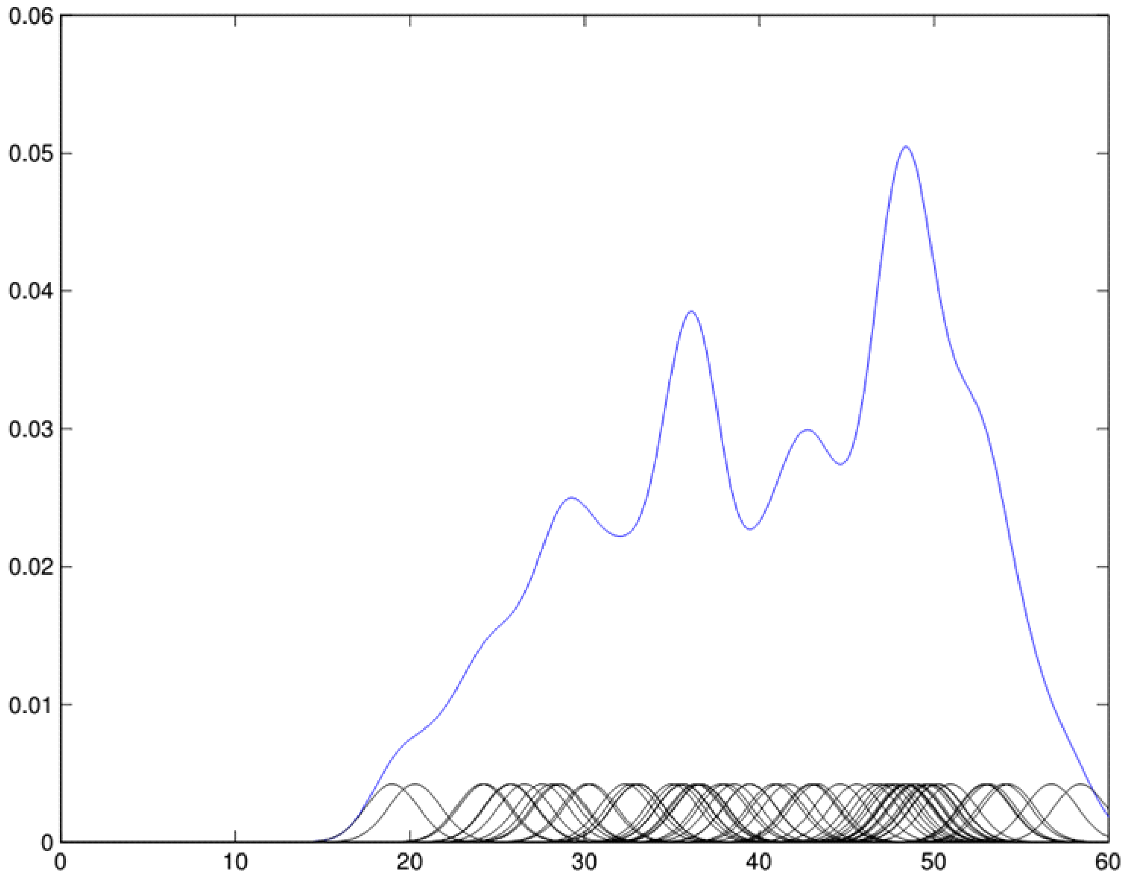
1. This notion of kernel is distinct from that of an inner product/positive definite kernel.

2. The KDE is sometimes called the Parzen window. It was originally proposed by Rosenblatt (1956) and Parzen (1962).

3. The KDE is clearly nonparametric.

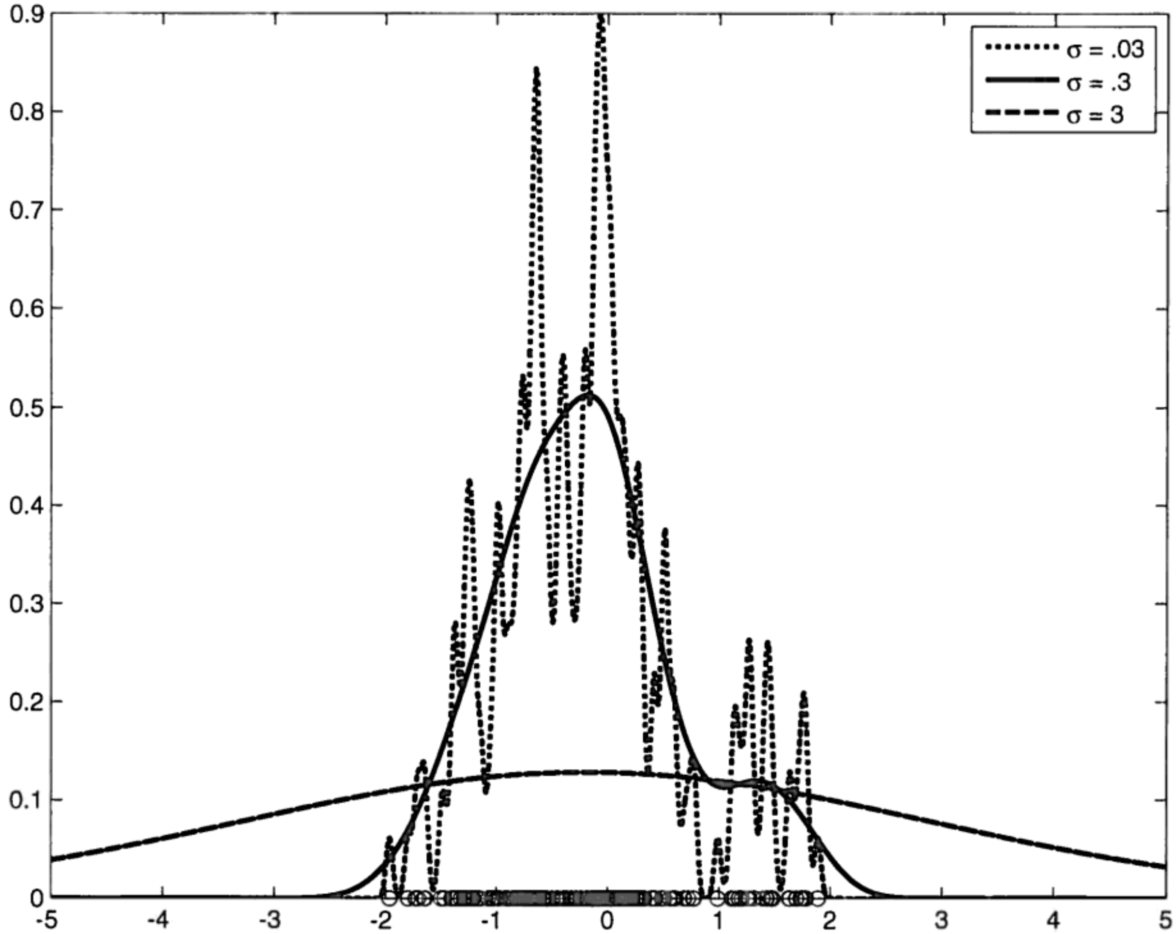
So why does it work? The KDE can be viewed as a superposition of shifted kernel functions. The more X_i in a given region of space, the more these shifted kernels accumulate.

Below is a KDE of midterm exam scores for a past version of EECS 545 (60 points max).



Model Selection

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



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

$$\mathrm{ISE}(\sigma) = \int \left(\hat{f}_\sigma(x) - f(x) \right)^2 dx$$

$$= \int \hat{f}_\sigma(x)^2 - 2 \int \hat{f}_\sigma(x) f(x)\, dx + \int f(x)^2 dx$$

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

$$\int \hat{f}_\sigma(x)^2 dx = \frac{1}{n^2} \sum_{i=1}^{n} \sum_{j=1}^{n} \int k_\sigma(x-X_i) k_\sigma(x-X_j) dx$$

$$= \frac{1}{n^2} \sum_i \sum_j k_{\sqrt{2}\sigma}(X_i - X_j),$$

since convolving Gaussian densities amounts to adding Gaussian RVs.

As for the third term,

$$\int \hat{f}_\sigma(x) f(x) dx = \mathbb{E}_{X\sim f}\left[\hat{f}_\sigma(X)\right]$$

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

$$\frac{1}{n} \sum \hat{f}(X_i)$$

$$\frac{1}{n} \sum \hat{f}_\sigma(X_i)$$

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

$$\frac{1}{n} \sum_{i=1}^{n} \hat{f}_\sigma^{-i}(X_i)$$

where

$$\hat{f}_\sigma^{-i}(x) = \frac{1}{n-1} \sum_{j \neq i} k_\sigma(x - X_j).$$

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

$$\hat{\sigma} = \arg\min_{\sigma} \sum_{i,j=1}^{n} k_{\sqrt{2}\sigma}(X_i, X_j) - \frac{2}{n(n-1)} \sum_{i=1}^{n} \sum_{j \neq i} k_\sigma(X_i, X_j)$$

The resulting procedure is called LS-LOOCV for least squares leave-one-out cross-validation.