# Kernel Regression

Shyue Ping Ong

University of California, San Diego

Fall 2022

# Overview

1. Preliminaries
2. k nearest neighbor
3. Kernel Density Estimation
4. Kernel Density Classification

# Preliminaries

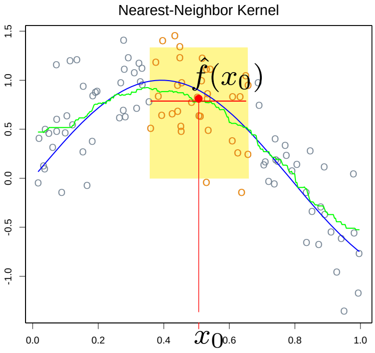
- Linear models, even those based on basis expansion, have high bias.
- In contrast, kernel methods fit many models to each point using the observations close to that point.
- Localization is based on a weighting function $K_\lambda(x_0; x_i)$ that assigns a weight to each observation $x_i$ based on distance to a query point.
- Typically, the kernel function has only a single parameter $(\lambda)$ to determine width of neighborhood.
- The “model” is the entire training data set.
- While undoubtedly effective in many instances, kernel methods lack interpretability that is often desired for scientific applications.

# $k$ Nearest Neighbor ($k$NN)

- Simplest possible model for prediction - even simpler than linear regression!
- Given a set of observations, we take the average of the $k$ nearest neighbors as an estimate.

$$E[Y|X=x] = \hat{f}(x) = Ave(y_i|x_i \in N_k(x))$$

- Prediction is bumpy, i.e., changes in average are discrete at the boundary between the inclusion and exclusion of a point.



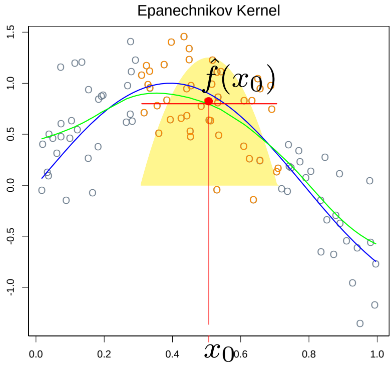
# Improving on *k*NN

- *k*NN gives equal weight to all points that falls within the *k* nearest neighbor region.
- Solution: use a weighted kernel that goes to zero smoothly with distance from point.
- Nadaraya-Watson kernel-weighted average:

$$\hat{f}(x) = \frac{\sum_{i=1}^{N} K_\lambda(x_0, x_i) y_i}{\sum_{i=1}^{N} K_\lambda(x_0, x_i)}$$

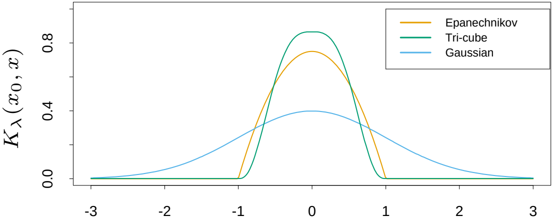
- Epanechnikov quadratic kernel:

$$K_\lambda(x_0, x) = D\left(\frac{|x - x_0|}{\lambda}\right), D(t) = \frac{3}{4}(1 - t^2) \text{ if } |t| \leq 1$$



# Considerations

- Smoothing parameter $\lambda$ determines the width of the local neighborhood. Large $\lambda$ means lower variance but higher bias.
- Metric window widths: As local density increases, vias decreases.
- Epanechnikov kernel is compact. Tri-cube kernel $D(t) = (1 - |t|^3)^3$ if $|t| \leq 1$ is another compact kernel that is flatter and differentiable at bounday.
- Gaussian kernel is a popular *non-compact* kernel. Standard deviation controls width of kernel.



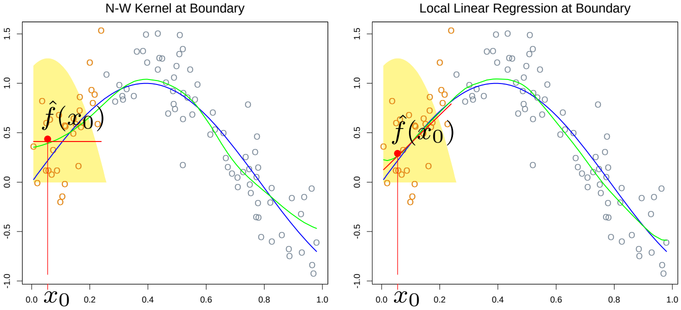
# Code

```
from sklearn.neighbors import KNeighborsRegressor
from sklearn.model_selection import cross_val_predict, KFold

kfold = KFold(n_splits=5, shuffle=True, random_state=42)
knn = KNeighborsRegressor(n_neighbors=14)
yhat_knn = cross_val_predict(knn, x, y, cv=kfold)
```

# Local linear/polynomial regression

- Local linear/polynomial regression can be used, which corrects bias at boundary regions at the expense of higher variance.
- For higher dimensions especially, local linear regression is preferred to local constant fit.



- Often used to interpolate within a region of feature space.

# Kernel Density Estimation

- Estimate the probability density function $\hat{f}_X(x)$ as:

$$\hat{f}_X(x_0) = \frac{\# x_i \in N(x_0)}{N\lambda}$$

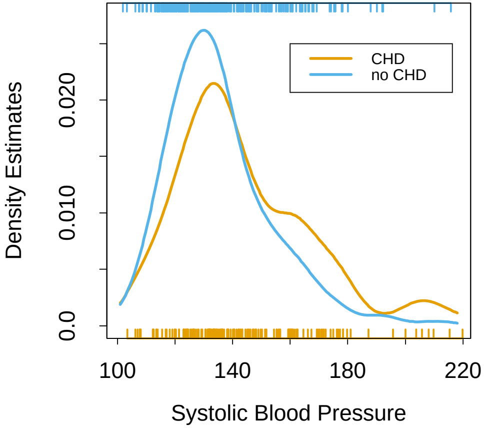
where $\lambda$ is the width of the bin and $N(x_0)$ is the neighbor of $x_0$ and $N$ is the total data count.

- Often, the smooth Parzen estimate is used.

$$\hat{f}_X(x_0) = \frac{1}{N\lambda} \sum_{i=1}^{N} K_\lambda(x_0, x_i)$$

- Popular choice of $K_\lambda$ is the Gaussian kernel $\phi(\frac{x - x_0}{\lambda})$.
- Essentially $f_X(x)$ is the convolution of the sample distribution with the Gaussian distribution with standard deviation $\lambda$.

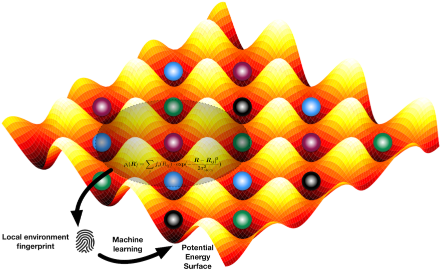
# Gaussian KDE



# Example of Gaussian Density Estimation in Interatomic Potentials

- Gaussian Approximation Potential[1] uses a smooth-overlap of atomic positions (SOAP) kernel in a Gaussian process model:

$$\rho_i(\boldsymbol{R}) = \sum_j f_c(R_{ij}) \cdot \exp(-\frac{|\boldsymbol{R} - \boldsymbol{R}_{ij}|^2}{2\sigma_{\text{atom}}^2}) = \sum_{nlm} c_{nlm}\, g_n(R) Y_{lm}(\hat{\boldsymbol{R}}),$$

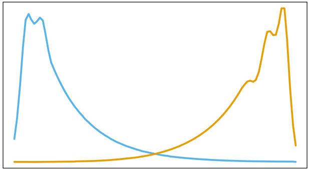

$\rho_i(\boldsymbol{R}) = \sum_j f_c(R_{ij}) \cdot \exp(-\frac{|\boldsymbol{R} - \boldsymbol{R}_{ij}|^2}{2\sigma_{\text{atom}}^2})$
Local environment fingerprint
Machine learning
Potential Energy Surface

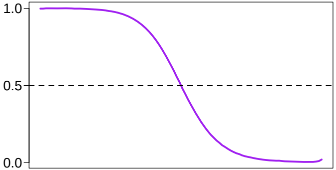
# Kernel Density Classification

- Given the kernel density estimate for each class $\hat{f}_j(X)$ and class prior $\pi_j$, we can use Bayes theorem to perform classification:

$$P(G = j|X = x_0) = \frac{\pi_j \hat{f}_j(x_0)}{\sum_{k=1}^{J} \pi_k \hat{f}_k(x_0)}$$

- However, density estimation for each class is not necessary if we only need to perform classification.
- The key is to estimate the posterior decision boundary between classes accurately.





# Naive Bayes

- Highly popular approach and often outperforms more sophisticated alternatives.
- Assumes features $X_k$ are independent, i.e., $f_j(X) = \prod_{k=1}^{p} f_j k(X_k)$, i.e., class conditional probabilities can be estimated using 1D kernel densities!

$$\begin{aligned}\log \frac{P(G = l|X)}{P(G = k|X)} &= \log \frac{\pi_l}{\pi_j} + \sum_{k=1}^{p} \log \frac{f_{lk}(X_k)}{f_{jk}(X_k)} \\ &= \alpha_l + \sum_{k=1}^{p} g_{lk}(X_k)\end{aligned}$$

We are converting a high-dimensional problem into simpler generalized additive model (see later lecture on GAMs).

# Radial Basis Functions

- Treat kernel functions as basis functions.

$$f(x) = \sum_{j=1}^{M} D(\frac{||x - \varepsilon_j||}{\lambda_j})\beta_j$$

- Each basis function is index by location ($\varepsilon_j$) and scale parameter $\lambda_j$.
- Gaussian function is a common choice for $D$.
- Parameters are optimized, typically using a least squares approach.

# Mixture Models

- Type of kernel model.

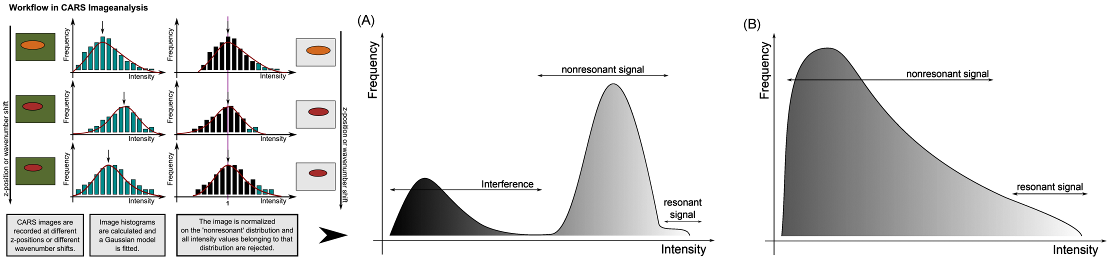
$$f(x) = \sum_{m=1}^{M} \alpha_m \phi(x; \mu_m, \mathbf{\Sigma_m})$$

- Again, Gaussian mixture model is by far the most common choice.
- If covariance matrices are constrained to be scalars. then it is similar to a radial basis expansion.
- Typically fitted using maximum likelihood approach / expectation maximization (next lecture).
- Probability that observation $i$ belongs in component $m$ is given by:

$$\hat{r}_{im} = \frac{\alpha_m \phi(x; \mu_m, \mathbf{\Sigma_m})}{\sum_{k=1}^{M} \alpha_k \phi(x; \mu_k, \mathbf{\Sigma_k})}$$

- Very often used in spectroscopy analysis.

# CARS spectroscopy analysis using Gaussian Mixtures



CARS images are recorded at different z-positions or different wavenumber shifts.

Image histograms are calculated and a Gaussian model is fitted.

The image is normalized on the 'nonresonant' distribution and all intensity values belonging to that distribution are rejected.

Figure: Coherent anti-Stokes Raman scattering (CARS) analysis. For rapid (online) determination of chemical composition. From ref. [2]

# Bibliography

- Albert P Bartók, Mike C Payne, Risi Kondor, and Gábor Csányi.
  Gaussian approximation potentials: The accuracy of quantum mechanics, without the electrons.
  *Physical Review Letters*, 104:136403, 2010.

- Nadine Vogler, Thomas Bocklitz, Melissa Mariani, Volker Deckert, Aneta Markova, Peter Schelkens, Petra Rösch, Denis Akimov, Benjamin Dietzek, and Jürgen Popp.
  Separation of CARS image contributions with a Gaussian mixture model.
  *Journal of the Optical Society of America A*, 27(6):1361, June 2010.

The End