Local averaging methods

MATH-412 - Statistical Machine Learning

Principle of local averaging methods

Goal : solve a (non-linear) regression problem.

Principle : At a point x predict the corresponding value y by a weighted mean of the y_i for x_i some neighbors of x.

Mathematically, we consider decision functions of the form :

$$\widehat{f} : x \mapsto \sum_{i=1}^{n} \omega_i(x)\, y_i$$

where $\omega_i(x)$ are weights that sum to 1, that depend on the input training data and that account for some form of similarity between the input x and the previously seen inputs x_i.

Idea : If points x_i with significant weights are closest to x and $f^*(x) = \mathbb{E}[Y|X = x]$ is continuous then $\widehat{f}$ should approximate f^* as n increases.

Some local averaging methods

- K-nearest neighbors
- Histogram based methods
- Nadaraya-Watson prediction functions (aka kernel smoothers)

K-nearest neighbors

Assume that $\mathcal{X}$ equipped with some distance d.

Let $V_k(x)$ the set of the k nearest neighbors of x for the distance d.

The weights are defined as :

$$\omega_i(x) = \frac{1_{\{x_i \in V_k(x)\}}}{k}.$$

The decision function is then

$$\widehat{f}(x) = \sum_{i=1}^{n} \omega_i(x)\, y_i$$

Histogram based methods

Relies on a finite or countable partition $\{A_1, A_2, \ldots\}$ of $\mathcal{X}$.

Let $s(x, x_i) = \sum_{k=1}^{K} 1_{\{x \in A_k\}} 1_{\{x_i \in A_k\}}$. So $s(x, x_i) = 1$ iff x and x_i are in the same bin.

Pick the weights :

$$\omega_i(x) = \tilde{s}(x, x_i) = \frac{s(x, x_i)}{\sum_{j=1}^{n} s(x, x_j)}$$

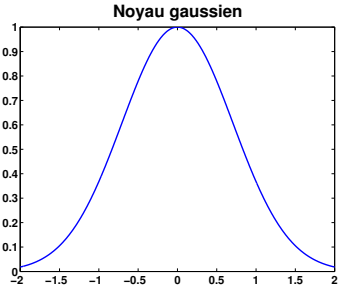
with the convention $\frac{0}{0} = 0$. The prediction function is then

$$\widehat{f}(x) = \sum_{i=1}^{n} \omega_i(x)\, y_i$$

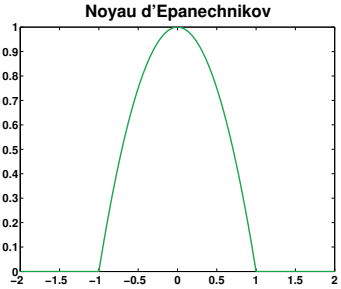
- Decision trees are actually histogram based methods, based on a partition that is learnt from the same data.

Convolution kernels

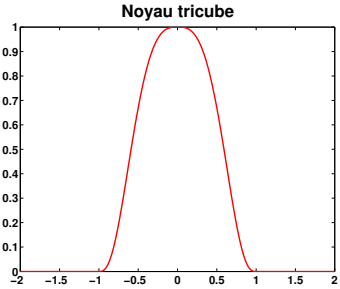
Convolution kernels are functions $K : \mathbb{R} \to \mathbb{R}_+$.



$t \mapsto \exp(-t^2)$



$t \mapsto (1-t^2)_+$



$t \mapsto (1-|t|^3)_+^3$

where $(x)_+ = \max(0, x)$ denotes the positive part.

- First used by Parzen and Rosenblatt for density estimation.
- Are naturally extended to $\mathbb{R}^p$ using $K_p : x \mapsto K(\|x\|)$.

Nadaraya-Watson estimators (aka kernel smoothers)

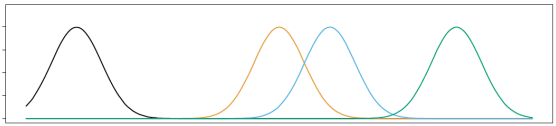
Principle : use the convolution kernels to define a similarity measure that depends on the Euclidean distance.

Weights take the form :

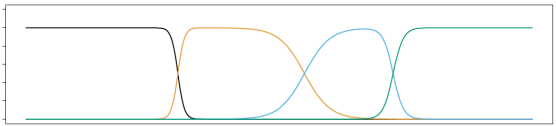
$$\omega_i(x) = \tilde{s}(x, x_i) = \frac{s(x, x_i)}{\sum_{j=1}^n s(x, x_j)} \quad \text{with} \quad s(x, x_i) = K\Big(\frac{\|x - x_i\|}{h}\Big).$$

$\rightarrow h$ is a *bandwidth hyperparameter* that control the scale.

$s(x, x_i) = K\Big(\frac{\|x - x_i\|}{h}\Big)$



$\tilde{s}(x, x_i)$



Star velocity estimation with kNN

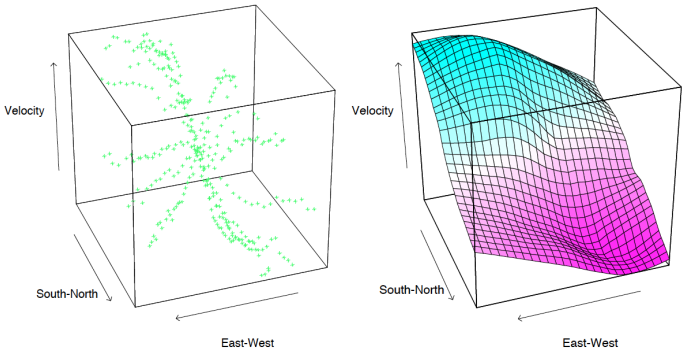


FIGURE 6.8. *The left panel shows three-dimensional data, where the response is the velocity measurements on a galaxy, and the two predictors record positions on the celestial sphere. The unusual "star"-shaped design indicates the way the measurements were made, and results in an extremely irregular boundary. The right panel shows the results of local linear regression smoothing in* $\mathbb{R}^2$*, using a nearest-neighbor window with* 15% *of the data.*

Comments on local averaging methods

- Methods from non-parametric statistics
- Generalizes to other similarity measures (i.e. $s(x,x) \geq s(x,z) \geq 0$).
- Their suffer seriously from the curse of dimensionality
- But k-NN is adaptive to the intrinsic dimensionality and scale of the data.
- They are a particular case of *local regression models* (degree 0).
- They are linear smoothers (aka linear estimator) :

$$\widehat{\boldsymbol{f}} = \widetilde{\mathbf{S}}\boldsymbol{y} \qquad \text{with} \qquad \widetilde{\mathbf{S}}_{i,j} = \tilde{s}(x_i, x_j),\ \widehat{\boldsymbol{f}} = \big(\widehat{f}(x_i)\big)_i,\ \text{and}\ \boldsymbol{y} = (y_i)_i$$

- If the similarity measure $s(x,z)$ does not depend on the data set [1] then the LOO risk estimate takes the form

$$\widehat{R}^{\text{LOO}} = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{y_i - \widehat{f}_{-i}(x_i)}{1 - \tilde{s}(x_i, x_i)}\right)^2$$

1. This assumption fails for k-nearest neighbor

Local Empirical Risk minimization

Idea : Solve a local version of the ERM by introducing weights $s(x, x_i) \geq 0$ that are large if x and x_i are close or similar. Solve

$$f_x = \arg\min_{f \in S} \frac{1}{n} \sum_{i=1}^{n} s(x, x_i)\, \ell(f(x_i), y_i)$$

and define $\widehat{f}(x) = f_x(x)$. In particular if the local prediction function is a constant prediction function $f_x(z) = a_x$ then

$$\widehat{f}(x) = a_x \quad \text{with} \quad a_x = \arg\min_{a \in \mathcal{A}} \frac{1}{n} \sum_{i=1}^{n} s(x, x_i)\, \ell(a, y_i)$$

Local averaging and local quadratic ERM

Consider the particular case of the square loss :

- $\mathcal{A} = \mathcal{Y} = \mathbb{R}$ and $\ell(a, y) = \frac{1}{2}(a - y)^2$
- with *constant local prediction functions* : $f(x') = a$

We need to solve :

$$a_x = \arg\min_a \frac{1}{2n} \sum_{i=1}^{n} s(x, x_i)\,(a - y_i)^2$$

Setting the gradient of the local ER to zero we get : $0 = a_x \sum_{i=1}^{n} s(x, x_i) - \sum_{i=1}^{n} s(x, x_i)\, y_i$

So that $\boxed{\widehat{f}(x) = a_x = \sum_{i=1}^{n} \tilde{s}(x, x_i)\, y_i}$ with $\tilde{s}(x, x_i) = \dfrac{s(x, x_i)}{\sum_{j=1}^{n} s(x, x_j)}$,

and we recover local averaging prediction functions.

Local linear regression

- We still consider $\ell(a, y) = \frac{1}{2}(a - y)^2$
- but now we consider local prediction functions that are linear functions : $f(\mathbf{x}') = \boldsymbol{w}^\top \mathbf{x}' + b$

$$\widehat{f}(\mathbf{x}) = \boldsymbol{w}_{\mathbf{x}}^\top \mathbf{x} + b_{\mathbf{x}} \quad \text{width} \quad (\boldsymbol{w}_{\mathbf{x}}, b_{\mathbf{x}}) = \arg\min_{\boldsymbol{w}, b} \frac{1}{2n} \sum_{i=1}^{n} s(\mathbf{x}, \mathbf{x}_i)(\boldsymbol{w}^\top \mathbf{x}_i + b - y_i)^2$$

- Local linear regression is also known as LOWESS (Locally weighted scattergram smoothing) or it generalization called LOESS.
- Local linear regression can generalized to *local polynomial regression* and *local spline regression*.