From Kernel Regression to Attention Mechanisms

A six decade journey (1960-2020)

Peyman Milanfar
Google

Kernel Regression
(Statistics)
1960-90s

Data-Adaptive Filters
(Signal Processing)
1990-2010s

Attention
(Machine Learning)
2010-2020s

<u>Kernel Regression/Smoothing</u> (Nadaraya-Watson, '64)

Objective: fit a nonlinear relationship to paired data

$$y_i = y(x_i)$$

At any position x, one can estimate:

$$\hat{y}(x) = \frac{\sum_i K(x, x_i) y_i}{\sum_i K(x, x_i)}$$

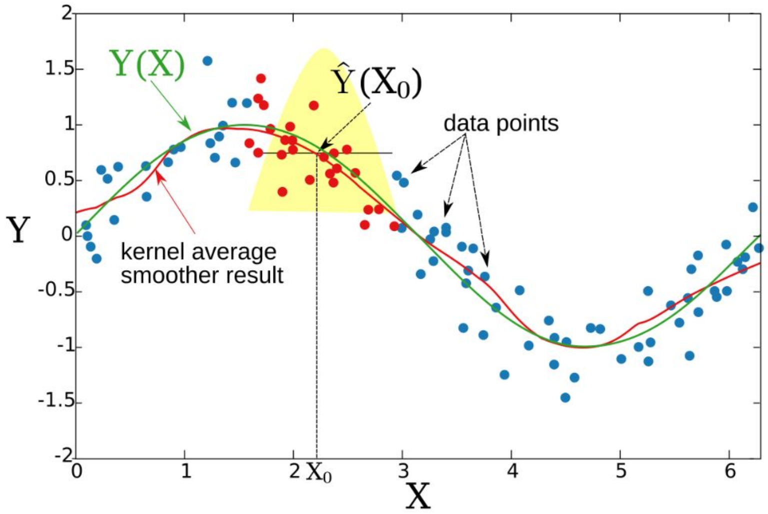
Positions

$$K(x_i, x_j) = \exp\left\{\frac{-\|x_i - x_j\|^2}{h_x^2}\right\}$$

<u>Kernel Regression/Smoothing</u> (Nadaraya-Watson, '64)

Positions

$$K(x_i, x_j) = \exp\left\{\frac{-\|x_i - x_j\|^2}{h_x^2}\right\}$$



Kernel Regression/Smoothing (Nadaraya-Watson, ‘64)

Positions

$$K(x_i, x_j) = \exp\left\{\frac{-\|x_i - x_j\|^2}{h_x^2}\right\}$$

Bilateral Filter (Tomasi, Manduchi, ‘98)

Positions

Pixels value

$$K(x_i, x_j, y_i, y_j) = \exp\left\{\frac{-\|x_i - x_j\|^2}{h_x^2} + \frac{-(y_i - y_j)^2}{h_y^2}\right\}$$

Bilateral Filter (Tomasi, Manduchi, '98)

Positions

Pixels value

$$K(x_i, x_j, y_i, y_j) = \exp\left\{ \frac{-\|x_i - x_j\|^2}{h_x^2} + \frac{-(y_i - y_j)^2}{h_y^2} \right\}$$

Non-local Means (Buades, Coll, Morel, '05)

Positions

Patches of Pixels

$$K(x_i, x_j, y_i, y_j) = \exp\left\{ \frac{-\|x_i - x_j\|^2}{h_x^2} + \frac{-\|\mathbf{y}_i - \mathbf{y}_j\|^2}{h_y^2} \right\}$$

∞

Non-local Means (Buades, Coll, Morel, '05)

Positions

Patches

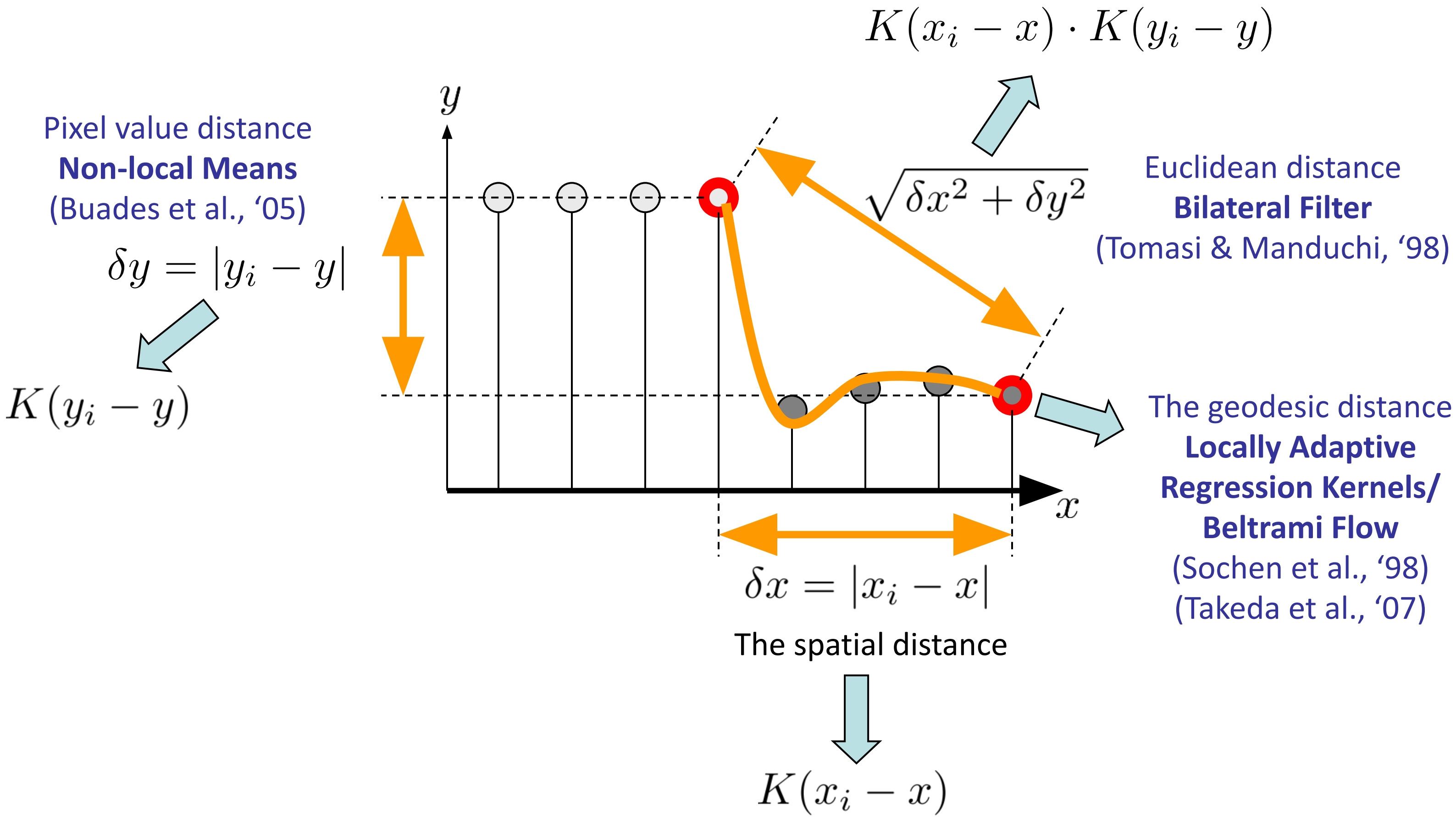
$$K(x_i, x_j, y_i, y_j) = \exp\left\{ \frac{-\|x_i - x_j\|^2}{h_x^2} + \frac{-\|\mathbf{y}_i - \mathbf{y}_j\|^2}{h_y^2} \right\}$$

∞

Locally Adaptive Regression Kernel (Takeda et al. '07)(Sochen et al., '98)

Learned Metric

$$K(x_i, x_j, y_i, y_j) = \exp\left\{ -\boxed{(x_i - x_j)^T \widehat{\mathbf{C}}_{ij}(y)(x_i - x_j)} \right\}$$


$K(x_i - x) \cdot K(y_i - y)$
y
Pixel value distance
Non-local Means
(Buades et al., ‘05)
$\delta y = |y_i - y|$
$K(y_i - y)$
$\sqrt{\delta x^2 + \delta y^2}$
Euclidean distance
Bilateral Filter
(Tomasi & Manduchi, ‘98)
The geodesic distance
Locally Adaptive Regression Kernels/ Beltrami Flow
(Sochen et al., ‘98)
(Takeda et al., ‘07)
x
$\delta x = |x_i - x|$
The spatial distance
$K(x_i - x)$

Consider Kernel with Augmented Variable:

$$\mathbf{t} = \begin{bmatrix} x \\ \mathbf{y} \end{bmatrix}$$

Positional "encoding"

Pixel Values

$$K(\mathbf{t}_i, \mathbf{t}_j) = \exp\left\{-(\mathbf{t}_i - \mathbf{t}_j)^T \mathbf{Q}_{i,j}(\mathbf{t}_i - \mathbf{t}_j)\right\}$$

$$\mathbf{Q} = \begin{bmatrix} \mathbf{Q}_x & 0 \\ 0 & \mathbf{Q}_y \end{bmatrix} \longleftarrow \text{Symmetric, positive-definite}$$

- Classical: $\mathbf{Q}_x = \frac{1}{h_x^2}\mathbf{I}$ and $\mathbf{Q}_{\mathbf{y}} = \mathbf{0}$
- Bilateral: $\mathbf{Q}_x = \frac{1}{h_x^2}\mathbf{I}$ and $\mathbf{Q}_{\mathbf{y}} = \frac{1}{h_y^2}\text{diag}[0, 0, \cdots, 1, \cdots, 0, 0]$
- Non-local Means: $\mathbf{Q}_{\mathbf{x}} = \mathbf{0}$ and $\mathbf{Q}_{\mathbf{y}} = \frac{1}{h_y^2}\mathbf{G}$
- LARK: $\mathbf{Q}_x = \mathbf{C}_{ij}$ and $\mathbf{Q}_{\mathbf{y}} = \mathbf{0}$.

P. Milanfar, "**A Tour of Modern Image Filtering**" IEEE Signal Processing Magazine, vol. 30, no. 1, pp. 106-128, **Jan. 2013**

1. Choose any kernel from a Reprod Kernel Hilbert Space

$$K(\mathbf{t}_i, \mathbf{t}_j) \quad = \quad \exp\left\{-(\mathbf{t}_i - \mathbf{t}_j)^T \mathbf{Q}_{i,j} (\mathbf{t}_i - \mathbf{t}_j)\right\}$$

2. Define **t** as a feature vector (latent space)

$$\mathbf{Q} = \begin{bmatrix} \mathbf{Q}_x & 0 \\ 0 & \mathbf{Q}_y \end{bmatrix}$$

3. Allow off-diagonal blocks and a more general factored form:

$$\mathbf{Q} = \mathbb{Q}^T \mathbb{K}$$

$$\tilde{\mathbf{y}} = \mathbb{V}\mathbf{t}$$

4. Allow output to have different dimensions than the input or the features

Attention

QUERY

KEY

VALUE

$$\sum_i \frac{e^{\langle \mathbb{Q}\mathbf{t}_i, \mathbb{K}\mathbf{t}_j \rangle}}{\sum_i e^{\langle \mathbb{Q}\mathbf{t}_i, \mathbb{K}\mathbf{t}_j \rangle}} \mathbb{V}\mathbf{t}_i$$