STAT 527: Assignment #3

Philip Pham

May 11, 2020

Problem 1

(a) *Proof.* We can rewrite

$$\hat{\beta} = \operatorname{argmin}_{\beta \in \mathbb{R}^{k+1}} \left(y - Z_{x_0}\beta\right)^\top W_{x_0} \left(y - Z_{x_0}\beta\right),$$

which is just the weighted least squares problem, so

$$\hat{\beta} = \left(Z_{x_0}^\top W_{x_0} Z_{x_0}\right)^{-1} Z_{x_0}^\top W_{x_0} y$$

Now, we can write $\hat{f}(x_0)$ as

$$\begin{aligned}\hat{f}(x_0) &= e_1^\top \hat{\beta} = e_1^\top \left(Z_{x_0}^\top W_{x_0} Z_{x_0}\right)^{-1} Z_{x_0}^\top W_{x_0} y \\ &= \left(e_1^\top \left(Z_{x_0}^\top W_{x_0} Z_{x_0}\right)^{-1} Z_{x_0}^\top W_{x_0}\right) y,\end{aligned}$$

so can write s_{x_0} as

$$s_{x_0} = W_{x_0} Z_{x_0} \left(Z_{x_0}^\top W_{x_0} Z_{x_0}\right)^{-1} e_1 \tag{1}$$

□

(b) See Listing 1 for a Python implementation. See Figure 1 for the plots of s_{x_0}.

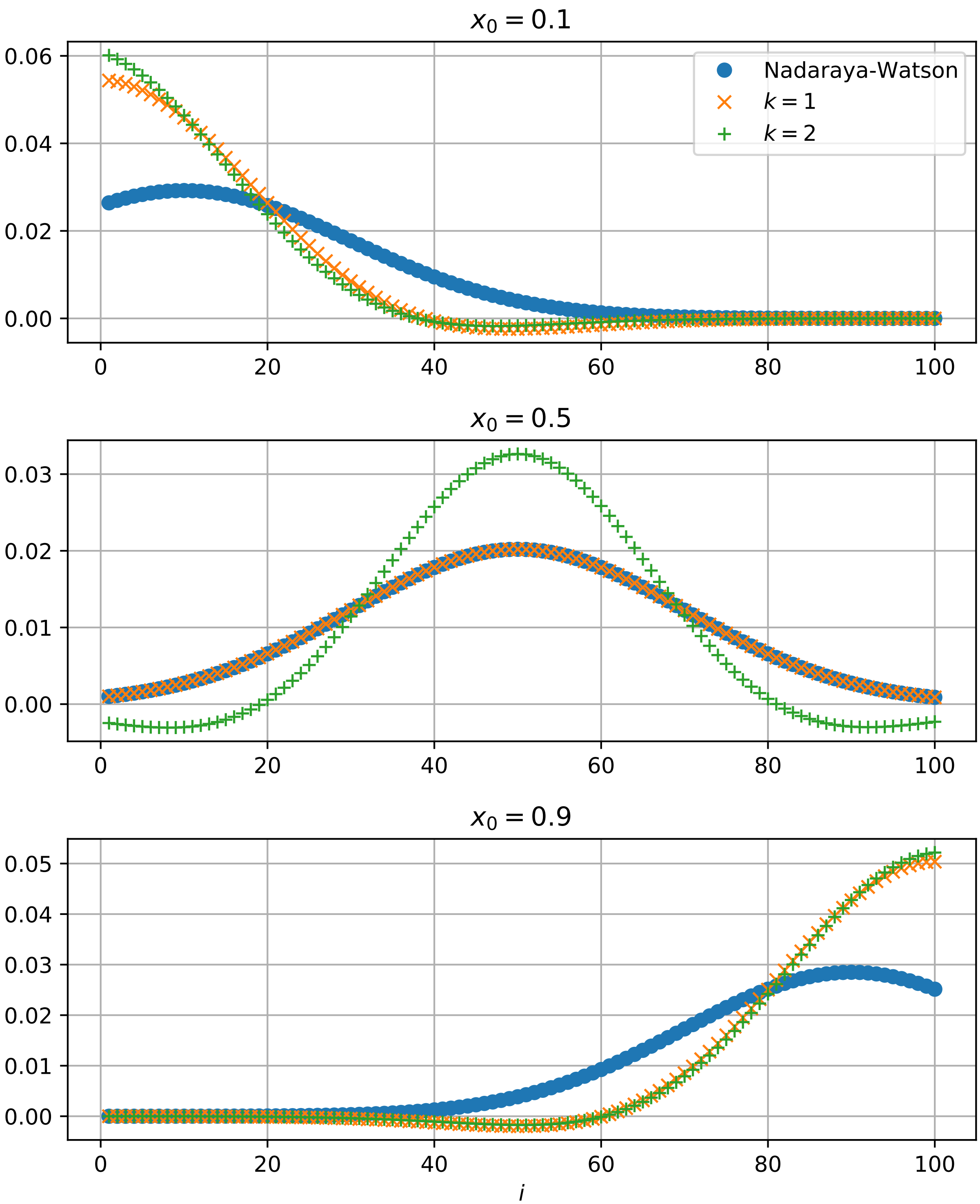
```
import numpy as np
from statsmodels.nonparametric import kernel_regression

def calculate_weights(n, h, k, x0):
    x = np.linspace(1/n, 1, n)
    Z = np.stack([np.power((x - x0)/h, i) for i in range(k + 1)], 1)
    W = np.diag(kernel_regression.kernel_func['gaussian'](h, x, x0))
    return W.dot(Z.dot(np.linalg.inv(Z.T.dot(W).dot(Z))))[:, 0]
```

Listing 1: A Python implementation of Equation 1.

In all cases, points observations associated with points near x_0 receive the most weight. Increasing k appears to more strongly emphasize the local neighborhood. The effect is more pronounced at $x_0 = 0.5$ for $k = 2$. At the tail and head, at $k = 1$ and $k = 2$ do not differ much.

Figure 1: Plots of s_{x_0} with various k.



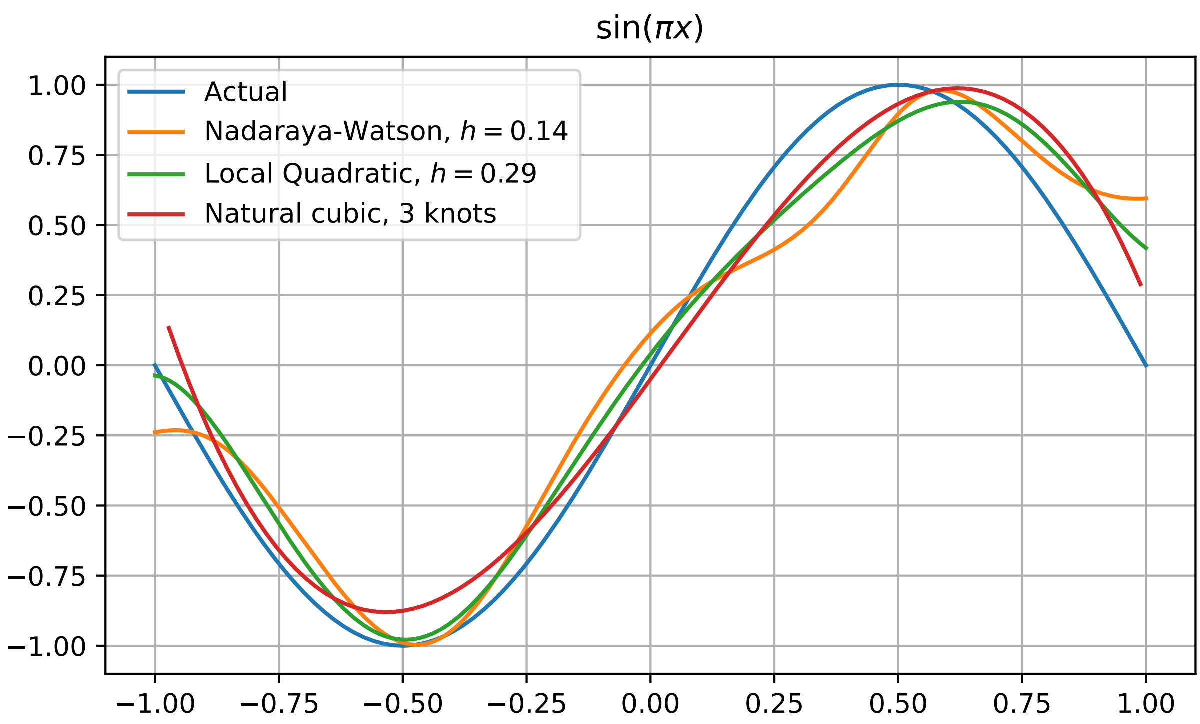


Figure 2: Estimates of $f(x) = \sin(\pi x)$.

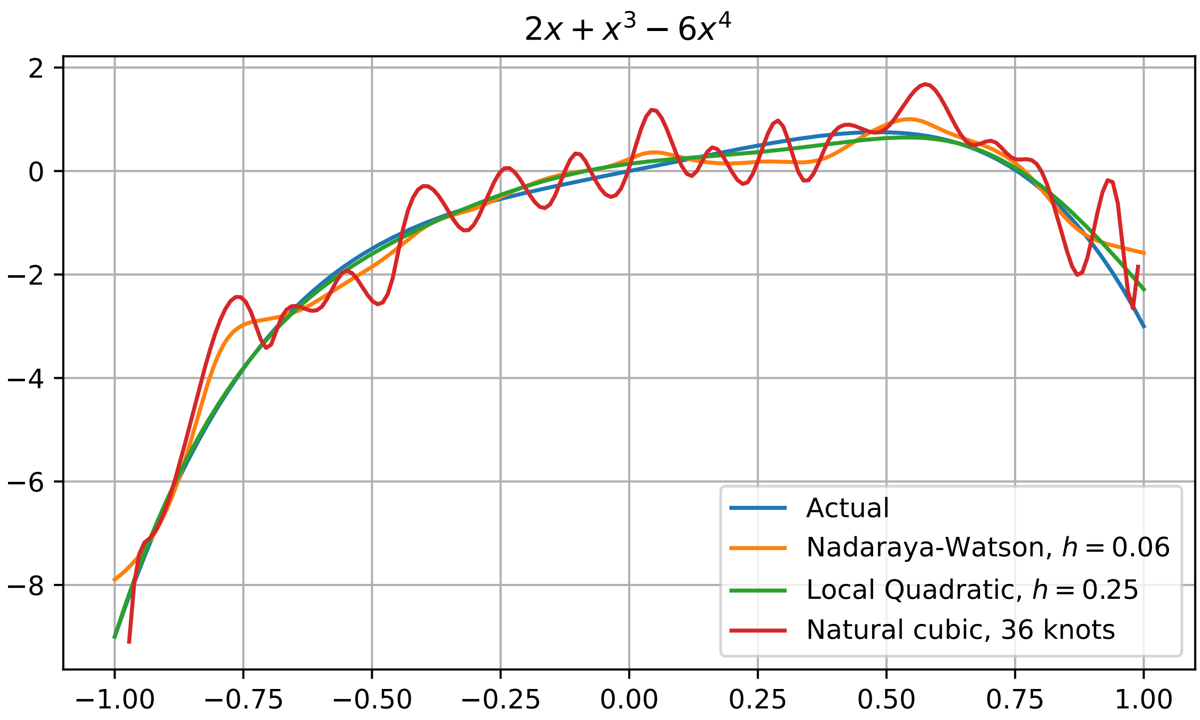


Figure 3: Estimates of $f(x) = 2x + x^6 - 6x^4$.

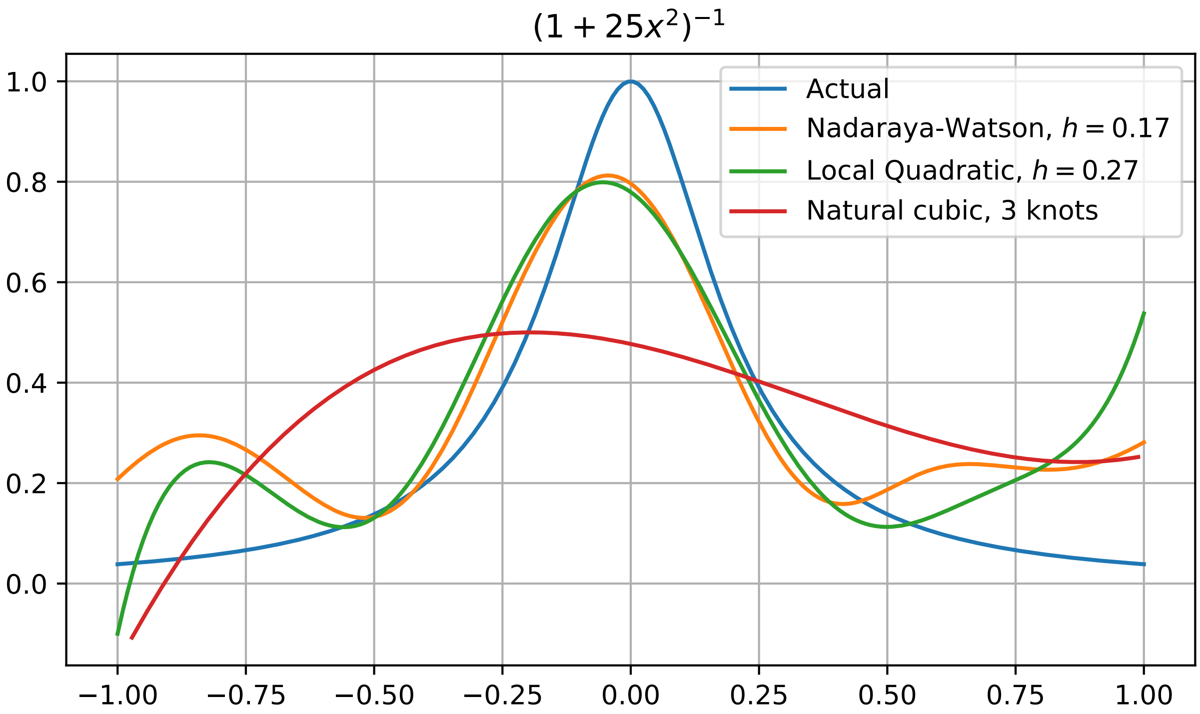


Figure 4: Estimates of $f(x) = \left(1 + 25x^2\right)^{-1}$.

Problem 2

(a) See Figure a.

(b) See Figure b.

(c) See Figure c.

All three estimators do well with $f(x) = \sin(\pi x)$. The Nadaraya-Watson estimator seems to have trouble with the tails.

For $f(x) = 2x + x^6 - 6x^4$, the local quadractic regression does best here. The spline method uses many knots and appears quite noisy in particular.

For $f(x) = \left(1 + 25x^2\right)^{-1}$, all three estimators struggle. The tails are badly modeled. The spline estimator does the worst and isn't able separate the noise from the true signal.