Lokal fitting

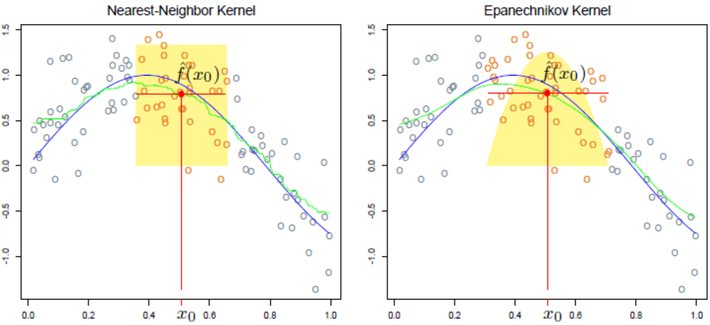
Peter N. Bakker

08-06-2023

Kerneudglatningsmetoder

- Kerneudglatningsmetoder anvender en kernefunktion, $K_\lambda(x_0, x_i)$, til at udglatte data når man estimerer sin funktion. Dvs. man vægter x_i ift. hvor langt væk de er fra x_0.
  - k-NN $\hat{f}(x) = \text{Ave}(y_i | x_i \in N_k(x))$.
  - Nadaraya-Watson, vægte aftager med afstand. $\hat{f}(x_0) = \frac{\Sigma K_\lambda(x_0,x_i) y_i}{\Sigma K_\lambda(x_0,x_i)}$.
  - $K_\lambda(x_0, x_i) = D\left(\frac{|x-x_0|}{\lambda}\right)$.
- λ er en hyperparameter (stort set den eneste) der bestemmer båndbredden og dermed styrer bias/varians.
- Kan også bestemmes som en adaptiv funktion $h_\lambda(x_0)$.
- Kerneudglatning giver desværre stort bias i endepunkterne grundet manglende datapunkter (til højre eller venstre).

Kernel smoothing illustration



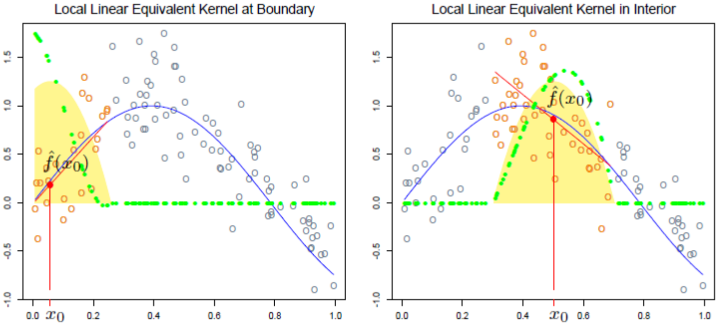
Lokal lineær regression

- Bias fra kerneudglatning kan minimeres ved *local linear regression*. Idéen er at fitte mere lineære led i yderpunkterne.
- Dette gøres ved at minimere en vægtet kvadrat sum:

$$\min_{\alpha(x_0),\beta(x_0)} \sum_{i=1}^{N} K_\lambda(x_0, x_i)[y_i - \alpha(x_0) - \beta(x_0)x_i]^2$$

- Løsning: $\hat{f}(x_0) = \hat{\alpha}(x_0) + \hat{\beta}(x_0)x_0$ (matrix-form $= \sum_{i=1}^{N} l_i(x_0)y_i$.
- Vægtene $l_i(x_0)$ kan kaldes en *ækvivalent kerne*.
- **Bemærk:** at vi kun fitter til x_0 (og gør det for alle x_i), bemærk at løsningen er lineær i y_i.
- Kan oversættet til $\mathcal{R}^p$ dimensioner. Afstanden findes nu som den euklidiske afstand. Er dog sjælden god i mere end 2 dimensioner.

Illustration



Bias

- Taylorudvikling af f giver:

$$
\begin{aligned}
E\hat{f}(x_0) &= \sum_{i=1}^{N} l_i(x_0) f(x_i) \\
&= f(x_0)\sum_{i=1}^{N} l_i(x_0) + f'(x_0)\sum_{i=1}^{N}(x_i - x_0) l_i(x_0) \\
&+ \frac{f''(x_0)}{2}\sum_{i=1}^{N}(x_i - x_0)^2 l_i(x_0) + R
\end{aligned}
$$

- Jf. Hastie et al. vises det at bias primært afhænger af $f''(x_0)$ og højere ordensled.
- Højere ordens ploynomer som 3. og 4. grads kan reducere bias (men koster selvfølgelig på varians.

Båndbreddevalg

- Båndbredde styrer som sagt bias/varians tradeoff i høj grad.
- $\hat{f} = S_\lambda y$ med $\{S_\lambda\}_{ij} = l_i(x_j)$.
- Valg af båndbredde kan påvirke udglatning og præcision.
- K-CV, GCV og LOOCV er oplagte at bruge her.
- De effektive frihedsgrader $\mathrm{tr} S_\lambda$ er det antal uafhængige parametre i modellen, der bidrager til modellens fleksibilitet og kompleksitet (benyttes f.eks. i GCV).

Lokal likelihood og tæthedsestimering

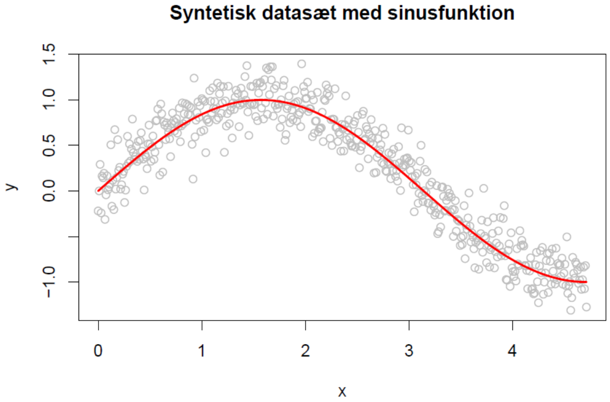
- Idéerne kan vægtes til vægtet likelihood. Og dermed kan enhver parametrisk model i princippet benytte sig af lokal fitting (f.eks. logistisk regression).

$$l(\beta(x_0)) = \sum_{i=1}^{N} K_\lambda(x_0, x_i) l(y_i, x_i^T \beta(x_0))$$

- Kan også bruges til estimere tæthedsfunktioner (Parzen estimator): $\hat{f}_X(x_0) = \frac{1}{N\lambda} \sum_{i=1}^{N} K_\lambda(x_0, x_i)$.
- Kan benyttes til Bayes klassifikation (*a posteriori* sandsynligheder): $\hat{P}(G = j | X = x_0) = \frac{\hat{\pi}_j \hat{f}_j(x_0)}{\sum_{k=1}^{J} \hat{\pi}_k \hat{f}_k(x_0)}$

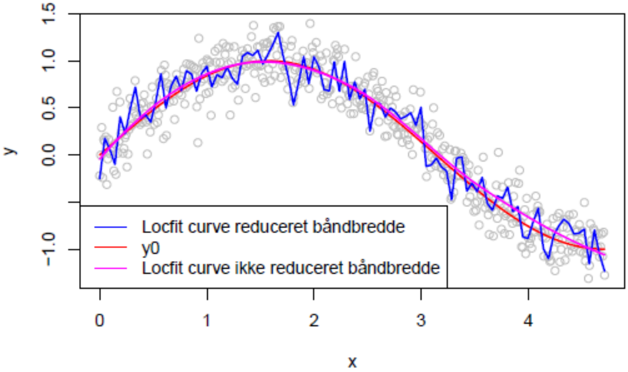
Dataeksempel: Syntetisk data

Sinus funktion med noget gaussisk støj.



Dataeksempel: Syntetisk data

- Jeg benytter lokal lineær regression (*locfit*) og viser for to forskellige båndbredder λ:



Dataeksempel: Syntetisk data

- Jeg benytter også CV og kan se hvordan valget af λ styrer CV-scoren (MSE).