5 Semi- und Nonparametrische Regression (I)

Aufgabe 1

Simulieren Sie $n = 250$ Daten gemäß dem folgenden, additiven Modell:

$$y_i = f(x_i) + \epsilon_i \quad \text{mit} \quad f(x) = \sin(2(4x-2)) + 2\exp(-(16)^2(x-0.5)^2) \quad \text{und} \quad \epsilon_i \overset{iid}{\sim} N(0, \sigma^2 = 0.3^2).$$

Die Einflussgröße x nimmt dabei feste Werte an, die gleichmäßig im Intervall $[0, 1]$ liegen. Das Ziel ist im Folgenden, die unbekannte Funktion $f(x)$ glatt zu schätzen. Da die wahre Gestalt von $f(x)$ unbekannt ist, soll als erster Schritt versucht werden, $f(x)$ durch ein Polynom vom Grad d in x darzustellen:
$f(x) = \mathcal{P}_d(x) = \sum_{l=0}^{d} \gamma_l x^l$.

(a) Fitten Sie dieses Modell für mehrere Werte von d und zeichnen Sie die daraus resultierende Schätzung sowie das wahre f in einen Scatterplot der Daten. Wie verändert sich die geschätzte Funktion für steigendes d? Berechnen Sie ausserdem für die verschiedenen Werte von d die Residuenquadratsumme dieses Polynomialschätzers auf den vorliegenden Daten. Ziehen Sie zudem $n_{test} = 500$ neue Testdaten und berechnen Sie die prädiktive Residuenquadratsumme, die man für diese Testdaten erhält. Was fällt auf?

(b) Unterteilen Sie den Wertebereich $[0, 1]$ von x in zehn gleichgroße Intervalle und fitten Sie auf jedem Intervall für die zu diesem Intervall gehörenden Beobachtungen ein kubisches Polynom. Vergleichen Sie das auf diese Art geschätzte stückweise Polynom mit den Ergebnissen aus a).

Die Funktion f soll nun durch B-Splines dargestellt werden im Sinne von

$$f(x) = \sum_{j=1}^{d} \gamma_j B_j(x),$$

wobei B_j die einzelnen Basisfunktionen bezeichnet und d ihre Anzahl. $\boldsymbol{B}(x_i) = (B_1(x_i), \ldots, B_d(x_i))'$ bezeichne den Vektor der Basisfunktionen für eine Beobachtung i und $\boldsymbol{\gamma} = (\gamma_1, \ldots, \gamma_d)'$ sei der Vektor der Gewichte der B-Splines.

(c) Leiten Sie für dieses Modell die Log-Likelihood, Scorefunktion und Fisher-Matrix von $\boldsymbol{\gamma}$ in Abhängigkeit von $\boldsymbol{B}(x_i)$ her.

(d) Mit der `R`-Funktion `bs()` aus dem Paket `splines` können in `R` B-Splines konstruiert werden. Berechnen und visualisieren Sie damit den B-Spline-Schätzer für verschiedene Splinegrade und verschiedene Anzahl an Knoten. Was geschieht, wenn Grad oder Knotenzahl steigen?

(e) Um das Problem der Wahl der Knotenzahl zu umgehen werden nun sogenannte P-Splines betrachet. Dabei sollen zweite Differenzen zwischen benachbarten Gewichten γ_j und γ_{j+1} bestraft werden. Wie wird der Schätzer des Gewichtsvektors $\boldsymbol{\gamma}$ hier berechnet?

(f) Schätzen Sie nun f durch P-Splines. Verwenden Sie dazu die Funktion `gam()` aus dem Paket `mgcv`.