

732G12 Data Mining

Föreläsning 10

Johan Alenlöv

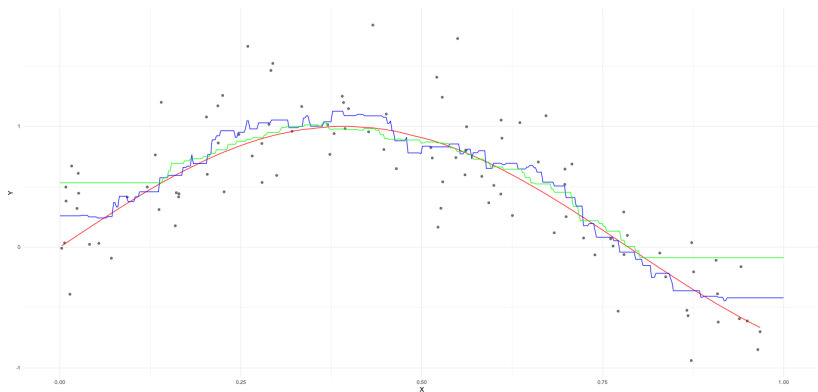
IDA, Linköping University, Sweden

- One-Dimensional Kernel Smoothers
- Lokal Regression
- Generaliserade Additiva Modeller
- Sammanfattning av Kursen

One-Dimensional Kernel Smoothers

Vi tänker oss tillbaka till KNN och dess användning vid regression.

$$\hat{g}(x) = \text{Medel}(y_i | x_i \in N_k(x)).$$



Blå har $K = 10$ och grön $K = 30$.

One-Dimensional Kernel Smoothers

- Problem med hopp i funktionen.
- Ett sätt att lösa det är att vikta om punkterna.

Vi vill estimerar $g(x_0)$ men istället för att ge alla närliggande punkterna samma vikt så använder vi

$$\hat{g}(x_0) = \frac{\sum_{i=1}^n K_\lambda(x_0, x_i) y_i}{\sum_{i=1}^n K_\lambda(x_0, x_i)},$$

där $K_\lambda(x_0, x_i)$ är en viktfunktion (Kernel).

One-Dimensional Kernel Smoothers

Kernel-funktionen är en täthet, vanligtvis given på formen

$$K_\lambda(x_0, x_i) = D\left(\frac{|x_0 - x_i|}{\lambda}\right)$$
$$D(t) \propto \begin{cases} (1 - |t|^p)^q, & \text{om } |t| \leq 1, \\ 0 & \text{annars.} \end{cases}$$

eller $D(t)$ är täthetsfunktionen för en standard normalfördelning. λ kallas för "bandwidth".

Triangel : $p = 1, q = 1$

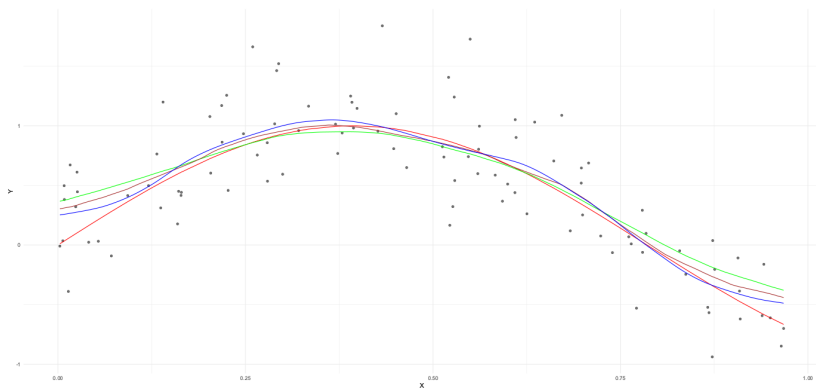
Epanechnikov : $p = 2, q = 1$

Quartic : $p = 2, q = 2$

Tri-cube : $p = 3, q = 3$

One-Dimensional Kernel Smoothers

Vi testar tre olika kernels (Triangel - Grön, Epanechnikov - Brun, Quartic - Blå) och får följande resultat



One-Dimensional Kernel Smoothers

- Ett sätt att göra "viktat KNN" som ger en "mjuk" funktion.
- Vanligtvis sätter vi inte ett antal grannar.
 - Låter λ styra hur stort fönster vi kollar i.
- Många olika val av kernels.
- Problem vid kanterna.

Vi har sedan tidigare delat upp variabelrummet i olika delar och gjort regression på varje del.

Istället för att dela upp rummet i förväg och göra regression i varje del, varför inte välja ett intervall runt den punkt där vi vill beräkna regressionen?

Hur välja område?

Inspirerat av det One-Dimensional Kernel Smoothers

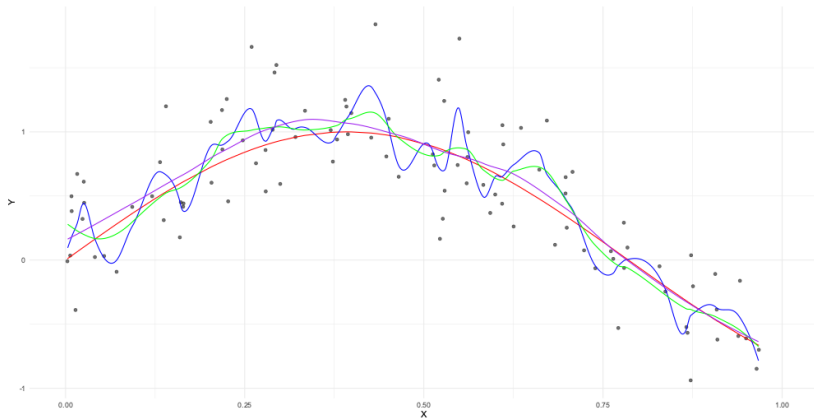
Gör (linjär) regression vid en punkt x_0 där vi viktar alla fel med en kernel.

För varje punkt x_0 där vi vill beräkna funktionen löser vi ett regressionsproblem

$$\min_{\beta(x_0)} \sum_{i=1}^n K_{\lambda}(x_0, x_i) [y_i - \beta_0(x_0) + \beta_1(x_0)x_i]^2.$$

Lokal Regression

Skattar lokal regression med olika λ med $p = q = 3$ (Tri-Cube)



- Kan såklart göra polynomregression istället för linjär regression.
- Går att göra för multiple regression,
 - Vanligt att man bara gör lokal regression för några enstaka parametrar.
 - Kan göra simultan lokal regression där vi skattar ett p -dimensionellt plan.
 - Funkar (generellt) dåligt för p större än 4.

Generaliserade Additiva Modeller

Allt det vi gjort denna och förra föreläsning handlar om att presentera flexibla modeller för att prediktera y givet x .

Nu ska vi använda detta för att skapa flexibla modeller för att prediktera y givet $x_1, x_2, x_3, \dots, x_p$.

Generaliserade Additiva Modeller (GAM) är en generell modell för att utöka linjär regression genom att tillåta icke-linjära funktioner av varje variabel, samtidigt som modellen är additiv.

Gemensamt för alla GAM modeller är att vi i vår regression eller klassificerings modell byter ut

$$\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_p x_p,$$

till

$$\beta_0 + \beta_1 g_1(x_1) + \beta_2 g_2(x_2) + \dots + \beta_p g_p(x_p),$$

där g_i är en ("mjuk") icke-linjär funktion.

För regression får vi följande typ av modell,

$$y_i = \beta_0 + \beta_1 g_1(x_1) + \beta_2 g_2(x_2) + \dots + \beta_p g_p(x_p) + \varepsilon_i,$$

där vi är fria att välja varje g_i själva.

T.ex. kan vi välja att några ska vara splines, någon kanske är en steg-funktion och någon är identitetsfunktionen.

För klassificering får vi följande typ av modell,

$$\log \left(\frac{p(x)}{1 - p(x)} \right) = \beta_0 + \beta_1 g_1(x_1) + \beta_2 g_2(x_2) + \dots + \beta_p g_p(x_p) + \varepsilon_i,$$

där vi är fria att välja varje g_i själva.

Här kan vi igen aktivt välja vilka typer av funktioner som passar bäst per variabel.

- GAMs låter oss anpassa en icke-linjär funktion för varje förklarande variabel.
- Eftersom modellen är additiv kan vi fortfarande undersöka effekten av varje förklarande variabel genom att hålla de andra konstanta.
- Hur "mjuka" varje funktion är kan vi beskriva genom frihetsgraderna.
- Den största nackdelen är att modellen måste vara additiv.
 - Detta gör att interaktioner mellan variabler missas.
 - Vi kan lägga till specifika interaktioner, men detta måste göras manuellt.

Sammanfattning av kursen

Sammanfattning i en mening:

- Givet data, hitta den bästa (mest lämpade), modellen som beskriver detta dataset.

Till vår hjälp har vi gått igenom väldigt många olika modeller och algoritmer.

- Modellval
 - Felfunktioner
 - Dela upp data i träning, validering, test.
 - Korsvalidering
 - AIC, BIC...
 - Variabelselektion
- Regularisering,
 - LASSO, Ridge

- Trädmodeller
 - Dela upp variabelrummet i rektanglar.
 - Varje rektangel får ett värde.
 - Olika regler för uppdelning beroende på problem.
- Beskärning av träd
 - Förbeskärning
 - Efterbeskärning
- Ensemblemetoder
 - Bagging - Använd bootstrap för att skapa många "oberoende" träd.
 - Random forest - Gör slumpmässiga ändringar i träden.
 - Boosting - Skapa många små träd, men modifiera datan mellan varje träd.

- K-närmaste grannar
 - Skattar värdet med hjälp av närmaste datapunkterna.
 - Kan förbättras genom att vikta med avståndet
- Naive bayes
 - Skatta klassificering med hjälp av bayes sats
 - För full fördelning krävs mycket data.
 - Görs ofta förenklingen att varje variabel är oberoende.

- Klusteranalys
 - Öövervakad inlärning.
 - K-means klustering
 - Hierarkisk klustering
 - K-medoid klustering
 - DBSCAN

- Icke-linjär regression
 - Grundidé är att hitta en transformation av förklarande variabler.
 - Gått igenom många olika transformationer.
- Basfunktioner
- Splines
- Kernelfunktioner
- Lokal regression
- Neurala nätverk
 - Olika typer av lager för olika problem.
 - Olika aktiveringsfunktioner.
 - Global approximation theorem.

Tack för att ni har lyssnat!

Nu är det bara projektet kvar.