LINKÖPING UNIVERSITY

732G12 Data Mining

Föreläsning 4

Josef Wilzén
IDA, Linköping University, Sweden

Dagens föreläsning

- K-närmaste grannar
- One-Dimensional Kernel Smoothers
- Lokal Regression
- GAM: Generaliserade Additativa Modeller
- Sammanfattning av Kursen

K-närmaste grannar

Idé basera predikation på de K datapunkter som är närmast.

Ger en icke-parametrisk metod för klassificering och regression.

Problem: Vad är närmast?

Avståndsmått

Vi behöver något som talar om för oss hur nära två datapunkter är.
Finns många alternativ som man kan välja, som ger olika resultat.

Euklidiskt avstånd

$$d(\mathbf{x}, \mathbf{y}) = \sqrt{\sum_{k=1}^{n} (x_k - y_k)^2}$$

Manhattan avstånd

$$d(\mathbf{x}, \mathbf{y}) = \sum_{k=1}^{n} |x_k - y_k|$$

K-närmaste grannar

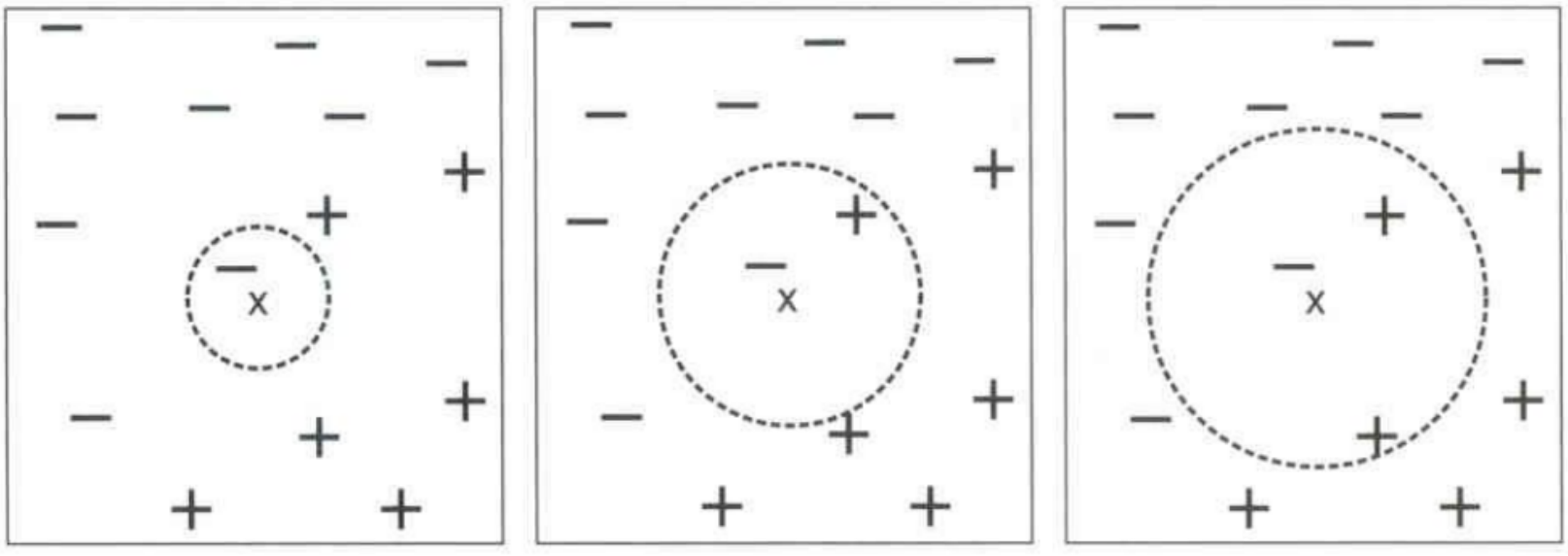
1. Låt k vara ditt valda antal grannar och D din träninigsdata.
2. För varje testdata $z = (\mathbf{x}', y') \in D$:
   - 2.1 Beräkna $d(\mathbf{x}, \mathbf{x}')$ (avstådet mellan z och all träningsdata)
   - 2.2 Välj $D_z \subseteq D$, de k närmaste träniningsdatan till z
   - 2.3 Låt $y' = \arg\max_v \sum_{(\mathbf{x}_i, y_i) \in D_z} \mathbf{I}_{v=y_i}$

2.3 är majoritetsvalet. Kan också vikta detta värde med avståndet:

2.3 $y' = \arg\max_v \sum_{(\mathbf{x}_i, y_i) \in D_z} w_i \mathbf{I}_{v=y_i}$.

För regression används medelvärde alternativt viktat medelvärde.

K-närmaste grannar



(a) 1-nearest neighbor (b) 2-nearest neighbor (c) 3-nearest neighbor

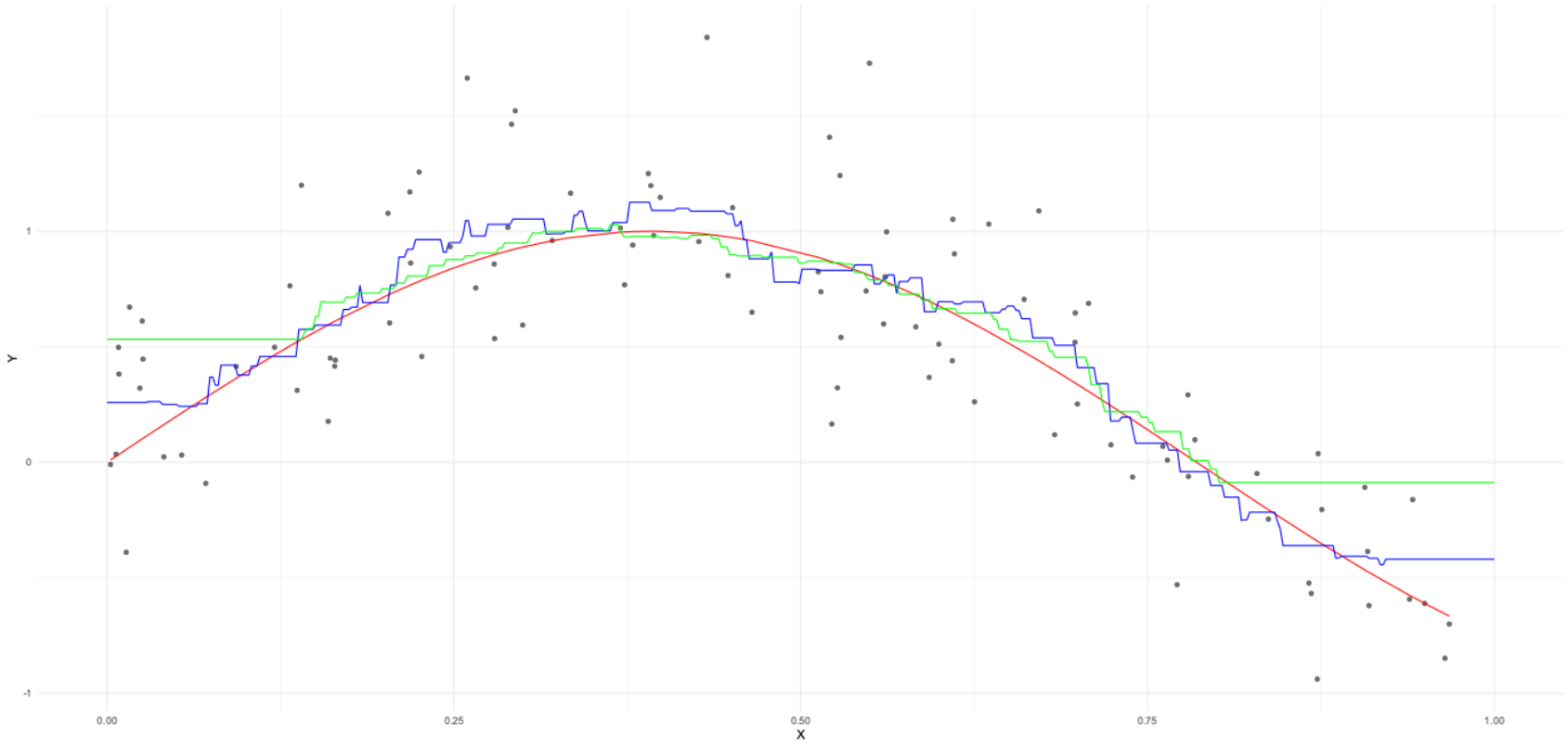
K-närmaste grannar

- Målet med modellen är att prediktera nya observationer
  - $\rightarrow$ enkel/lat metod
- Avståndsmått påverkar, påverkas stort av olika skalor på variabler
- Långsam anpassning.
- Känslig mot brus.
- Val av K har stor betydelse!
  - Litet K ger överanpassning.
  - Stort K ger underanpassning.
  - Korsvalidering kan användas för att bestämma K.
- Producerar godtyckligt utformade beslutsgränser.
- Problem i högre dimensioner.

K-närmaste grannar: exempel

Regressionsdata: röda linjen + brus

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



KNN: Blå har $K = 10$ och grön har $K = 30$.

One-Dimensional Kernel Smoothers

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

Vi vill estimera $g(x_0)$ 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 : $p = 1, q = 1$

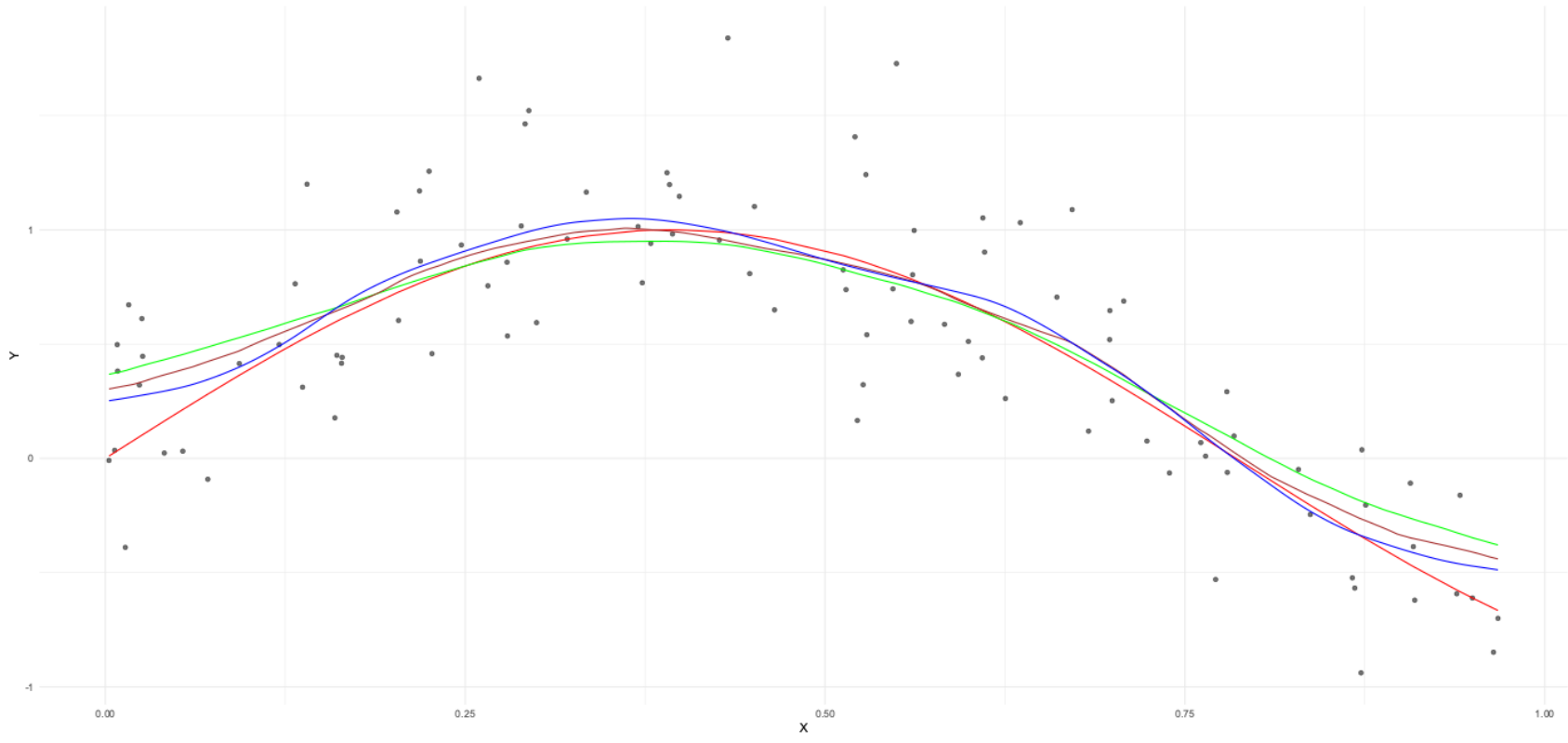
Epanechnikov : $p = 2, q = 1$

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.

Lokal Regression

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?

Lokal Regression

Inspirerat av det One-Dimensio Kenel Smoothers

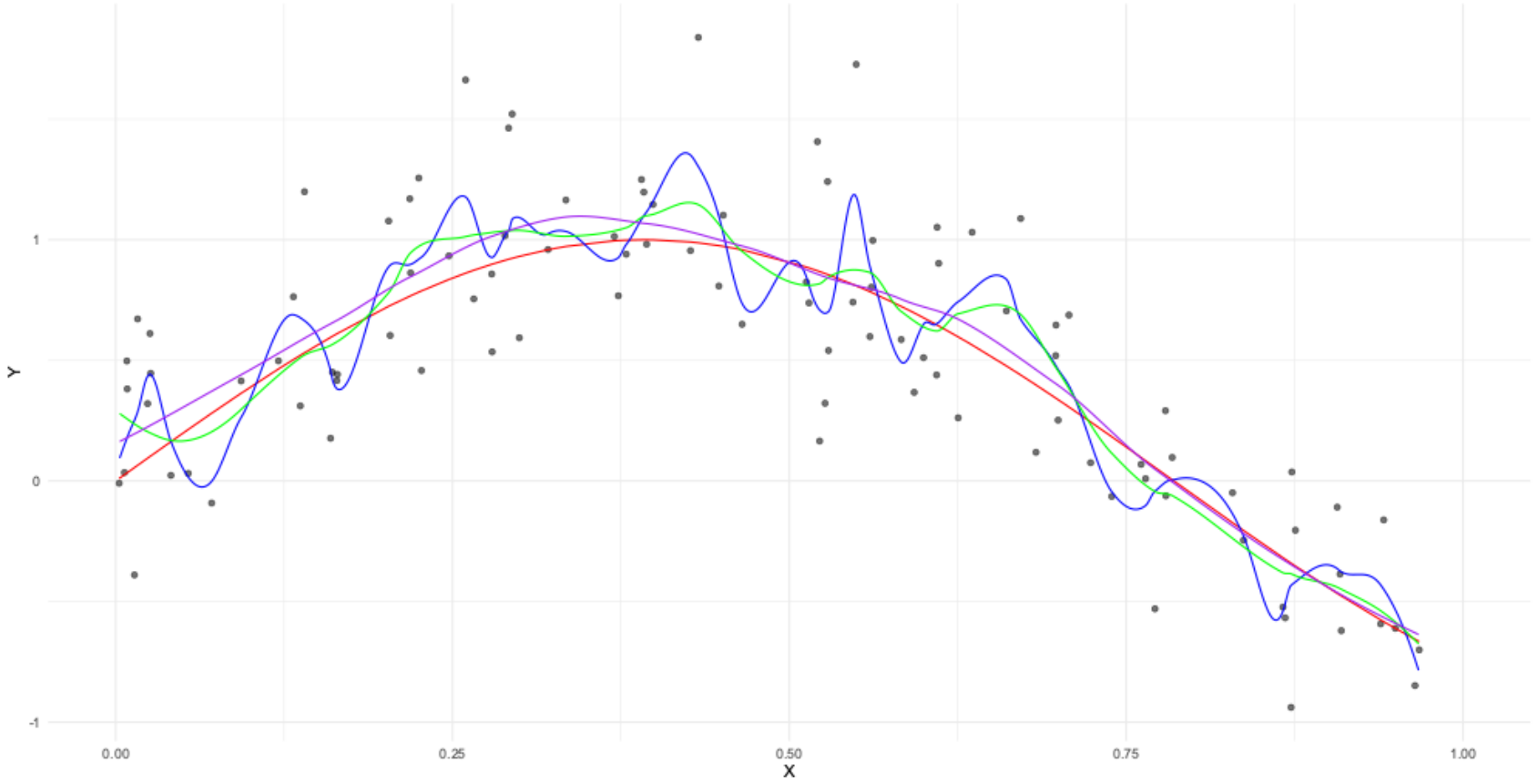
Gör (linjär) regression vid en punkt x_0 där vi viktar alla fel med en kernel.

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)



Lokal Regression

- Kan såklart göra polynomregression istället för linjär regression.
- Går att göra för multipl 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 Additativa Modeller

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 $x_1, x_2, x_3, \ldots, x_p$.

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

Generaliserade Additativa Modeller

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

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

till

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

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

Generaliserade Additativa Modeller

För regression får vi följande typ av modell,

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

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.

Generaliserade Additativa Modeller

För klassificering får vi följande typ av modell,

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

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.

Generaliserade Additativa Modeller

- GAMs låter oss anpassa en icke-linjär funktion för varje förklarande variabel.
- Eftersom modellen är additativ 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 additativ.
  - Detta gör att interaktioner mellan variabler missas.
  - Vi kan lägga till specifika interaktioner, men detta måste göras manuellt.