Bayesiansk statistik, 732g43, 7.5 hp

Moment 3 - Överanpassade modeller, regularisering, informationskriterium, modelljämförelse, Markov chain Monte Carlo (MCMC)

Bertil Wegmann

STIMA, IDA, Linköpings universitet

Översikt moment 3

- Över- och underanpassade modeller
- Informationsteori
- Regulariserande prior
- Informationskriterium
- Modelljämförelse
- Markov chain Monte Carlo (MCMC)

Overfitting och Underfitting

- **Overfitting**: en modell överanpassar data genom att skräddarsy modellen endast till befintliga data. Detta ger ofta dålig prediktionsförmåga på nya data, eftersom **modellen har lärt sig för mycket från data**.
  - Exempel: förklaringsgraden R^2 i multipel linjär regression ökar alltid för nya förklaringsvariabler, även gällande variabler som är helt orelaterade till den beroende variabeln.
- **Underfitting**: en modell underanpassar data genom att lära sig för lite från data. Detta ger också dålig prediktionsförmåga på nya data, eftersom **modellen är för enkel för att anpassa data**.
  - Exempel: relevanta förklaringsvariabler för multipel linjär regressionsmodell tas inte med i modellen.

Balanserade modeller

- **Två approacher** för att navigera mellan överanpassade och underanpassade modeller:
  - **Regulariserande prior**: modellen ska inte bli för exalterad över data, men heller inte vara för pessimistisk om informationen från data ("icke-Bayesianskt: penalized likelihood").
  - **Informationskriterium**: skattar den prediktiva förmågan för en modell. Bygger på **informationsteori**. Kända informationskriterium: **AIC**, **DIC**, **WAIC**.
- **Övergripande mål**: använda approacherna för att konstruera och kritisera modeller till att bli mer effektiva.
- **Kom dock ihåg**: alla modeller är fel, men vissa är bättre att använda än andra.

Exempel: över- och underanpassade modeller

- Linjär regressionsanalys med
  $y =$ liter per mil för en bils bränsleförbrukning
  $x =$ vikt i ton för en bil, standardiserad variabel

- Polynomisk linjär regressionsmodell:

$$Y_1, ..., Y_n | \mu_i, \sigma^2 \overset{iid}{\sim} N(\mu_i, \sigma^2)$$
$$\mu_i = \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \cdots + \beta_k x_i^k,$$

  där k är graden av den polynomiska regressionen.

- Oberoende priors för $(\beta_0, \beta_j, \ln \sigma)$:

$$\beta_0 \sim N(1, 1)$$
$$\beta_j \sim N(0, 5)$$
$$\ln \sigma \sim N(0, 2),$$

  där $j = 1, \ldots, k$.

Exempel: underanpassad modell #R kod 3.1

- Linjär regressionsanalys med den beroende variabeln $y = \text{liter per mil}$ för en bils bränsleförbrukning och inga förklaringsvariabler.

- Polynomisk linjär regressionsmodell av ordning 0:

$$Y_1, ..., Y_n | \mu_i, \sigma^2 \overset{iid}{\sim} N(\mu_i, \sigma^2)$$
$$\mu_i = \beta_0$$

- Oberoende priors för $(\beta_0, \ln \sigma)$:

$$\beta_0 \sim N(1, 1)$$

$$\ln \sigma \sim N(0, 2)$$

Exempel: överanpassad modell #R kod 3.2

- Linjär regressionsanalys med
  $y =$ liter per mil för en bils bränsleförbrukning
  $x =$ vikt i ton för en bil, standardiserad variabel

- Polynomisk linjär regressionsmodell av ordning 8:

$$Y_1, ..., Y_n | \mu_i, \sigma^2 \overset{iid}{\sim} N(\mu_i, \sigma^2)$$
$$\mu_i = \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \cdots + \beta_8 x_i^8$$

- Oberoende priors för $(\beta_0, \beta_j, \ln \sigma)$:

$$\beta_0 \sim N(1, 1)$$
$$\beta_j \sim N(0, 5)$$
$$\ln \sigma \sim N(0, 2),$$

  där $j = 1, \ldots, 8$.

Exempel: mer balanserad modell #R kod 3.3

- Linjär regressionsanalys med
  $y =$ liter per mil för en bils bränsleförbrukning
  $x =$ vikt i ton för en bil, standardiserad variabel

- Polynomisk linjär regressionsmodell av ordning 1:

$$Y_1, ..., Y_n | \mu_i, \sigma^2 \overset{iid}{\sim} N(\mu_i, \sigma^2)$$
$$\mu_i = \beta_0 + \beta_1 x_i$$

- Oberoende priors för $(\beta_0, \beta_1, \ln \sigma)$:

$$\beta_0 \sim N(1, 1)$$
$$\beta_1 \sim N(0, 5)$$
$$\ln \sigma \sim N(0, 2),$$

Informationsteori

- Ligger till grund för modellutvärdering via **regulariserande prior** och/eller **informationskriterium**.
- Informationsteori kan användas för att mäta avståndet för en modells anpassning till den perfekta anpassningen.
- Den bästa modellen är den modell som maximerar sannolikheten för data.
- Måttet **deviance** baseras på modellens sannolikhetsfördelning för data och är ett mått på den relativa osäkerheten i modellen jämfört med perfekt anpassning.
- Fokus är att få så liten **out-of-sample deviance** som möjligt.

K-L avvikelse och Deviance

- **Kullback-Leibler (K-L) avvikelse** kan användas för att mäta en modells avvikelse till perfekt anpassning.

- **K-L avvikelse** är den genomsnittliga skillnaden i log av sannolikheten mellan den perfekta modellen och modellen själv.

- **K-L avvikelsen** ökar ju längre modellen kommer från den perfekta modellen. Den fördelning för data eller modell som ger den lägsta avvikelsen är närmast den perfekta modellen. Jämförelse av modellers korrekthet kan utföras.

- En modells **Deviance** är en approximation av **K-L avvikelsen** och beräknas enligt:

$$D\left(q\right) = -2\sum_{i}\log\left(q_i\right),$$

där q_i är sannolikheten för varje observation i data.

Beräkning av Deviance #R kod 3.4

- Använd MAP skattningarna för dom linjära regressionsanalyserna tidigare, där den beroende variabeln $y = \text{liter per mil}$ för en bils bränsleförbrukning och $x =$ vikt i ton för en bil, standardiserad variabel.
- Beräkna $D(q)$ för varje modell.
- $D(q)$ blir lägst för den polynomiska regressionen av grad 8 i analogi med ökad förklaringsgrad R^2 för en mer komplicerad modell med fler antal förklaringsvariabler.
- Dela upp data i ett **training sample** och ett **test sample**. Första delen av datamaterialet används till **training sample**, $n_{train} = 20$, och andra delen används till **test sample**, $n_{test} = 12$, dvs totalt antal observationer är $n = 32$.
- Beräkna nu Deviance för respektive **training sample** och **test sample** och jämför mellan modellerna.

Regulariserande prior

- Regularisering motverkar överanpassning av data med hjälp av "skeptiska" priors.
- Regulariserande priorn bromsar in lärandet från data.
- Exempel: tighta priors kring 0 sätts på respektive parameterlutning i en linjär regressionsmodell.
- Om priorn tunas bra så motverkas överanpassning samtidigt som modellen lär sig huvuddragen från data.
- Obs! För mycket tuning med en för tight prior kan resultera i en underanpassning av data.

Regulariserande prior

- Hur tighta dessa "skeptiska" priors ska vara beror på data och modellen.
- **Deviance** för **training sample** blir alltid värre/högre med regulariserande priors.
- **Deviance** för **test sample** blir ofta bättre/lägre med regulariserande priors om man inte gjort priorn för tight för att underanpassa data.
- Bra tuning av den regulariserande priorn kan minska överanpassning av data rejält.
- Korsvalidering: dela upp data i **training sample** och **test sample** och välj den regulariserade prior som ger lägst **Deviance** på **test sample**.
- Obs! Om all data måste användas för att **träna** modellen så kan det vara svårt att tuna priorn på ett bra sätt.

Exempel: polynomisk regression med regulariserande prior

- Linjär regressionsanalys med
  $y =$ kilometer per liter för en bils bränsleförbrukning
  $x =$ vikt i ton för en bil, standardiserad variabel
  - Polynomisk linjär regressionsmodell av ordning 8:

$$Y_1, ..., Y_n | \mu_i, \sigma^2 \overset{iid}{\sim} N(\mu_i, \sigma^2)$$
$$\mu_i = \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \cdots + \beta_8 x_i^8$$

- Regulariserad prior på β_j **#R kod 3.5**:

$$\beta_0 \sim N(1, 1)$$

$$\beta_j \sim N(0, 0.1)$$

$$\ln \sigma \sim N(0, 2),$$

där $j = 1, \ldots, 8$.

Informationskriterium

- Vanliga informationskriterium för **out-of-sample deviance (testing deviance):**
  - **Akaike Information Criterium (AIC)**
  - **Deviance Information Criterium (DIC)**
  - **Widely Applicable Information Criterion (WAIC)**

- AIC, DIC och WAIC är alla mått på modellens prediktiva förmåga (out-of-sample deviance) och kan nominera den modell som gör bäst prediktioner.

AIC

- AIC är en skattning av den genomsnittliga out-of-sample deviance:

$$AIC = D_{train} + 2p,$$

där p är antalet parametrar som ska skattas i modellen och $2p \approx D_{out} - D_{train}$.

- AIC är en approximation av modellens prediktiva förmåga och är trovärdig om följande gäller:

1. flacka priors
2. posteriorn är approximativt multivariat normalfördelad
3. antalet observationer N är mycket större än antalet parametrar p

DIC

- Mer generellt mått än AIC, eftersom det inte kräver flacka priors.
- Bayesianskt informationskriterium, eftersom det beräknas från posteriorfördelningen för D_{train}, d.v.s $D_{train,s}$ för varje samplat värde s från posteriorn.
- Låt D vara posteriorfördelningen för Deviance, $\bar{D}$ det genomsnittliga värdet av D och $\hat{D}$ värdet på deviance för posteriormedelvärdet av parametrarna. Då gäller:

$$DIC = \bar{D} + (\bar{D} - \hat{D}) = \bar{D} + p_D = \hat{D} + 2p_D,$$

  där p_D är i analogi med antalet parametrar i modellen för att räkna ut AIC.
- Högre värde på p_D ger en mer flexibel modell för att skatta training sample, vilket ökar risken för överanpassning. Därför kallas p_D även för "penalty term".

WAIC

- Mer generellt mått än AIC och DIC, eftersom det inte kräver något av de antaganden som gäller för AIC och DIC. Kräver dock att observationerna i data är oberoende.
- Prediktionens osäkerhet mäts punktvis: observation för observation. Användbart, eftersom vissa observationer är svårare att prediktera än andra.
- WAIC kan delas upp i 2 delar.
- Första delen av WAIC: $Pr\left(y_i\right)$ är den genomsnittliga likelihooden för varje observation i över posteriorfördelningen och

$$lppd = \sum_{i=1}^{N} \log Pr\left(y_i\right)$$

står för *log-pointwise-predictive-density*.

WAIC

- Andra delen av WAIC: Antalet "effektiva" parametrar ges som

$$p_{WAIC} = \sum_{i=1}^{N} V\left(y_i\right),$$

där $V\left(y_i\right)$ är variansen av log-likelihooden av y_i från varje posteriordragning.

- Första och andra delen ger

$$WAIC = -2\left(lppd - p_{WAIC}\right) = D_{WAIC} + 2p_{WAIC},$$

där $D_{WAIC} = -2\,lppd$ är analogt med deviance D, men där sannolikheten för varje observation i data i stället är den genomsnittliga likelihooden för varje observation i över posteriorfördelningen.

Exempel: Beräkna och jämför AIC, DIC, WAIC #R kod 3.6

- Linjär regressionsanalys med
  $y =$ liter per mil för en bils bränsleförbrukning
  $x =$ vikt i ton för en bil, standardiserad variabel

- Jämför linjära regressionsmodeller med $k = 0, 1, 8$:

$$Y_1, ..., Y_n | \mu_i, \sigma^2 \overset{iid}{\sim} N(\mu_i, \sigma^2)$$
$$\mu_i = \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \cdots + \beta_k x_i^k,$$

  där k är graden av den polynomiska regressionen.

- Oberoende priors för $(\beta_0, \beta_j, \ln \sigma)$:

$$\beta_0 \sim N(1, 1)$$
$$\beta_j \sim N(0, 5)$$
$$\ln \sigma \sim N(0, 2),$$

  där $j = 1, \ldots, k$.

Modelljämförelse vs Modellval

- Modellval utifrån informationskriterier baseras på att välja den modell som ger lägst AIC/DIC/WAIC.
- Modellval tar inte hänsyn till information om den relativa prediktiva förmågan bland konkurrerande modeller.
- Modelljämförelse och genomsnittlig prediktiv förmåga över konkurrerande modeller tar hänsyn till små eller stora skillnader i prediktiva förmåga.
- Modelljämförelse: utvärdera modellerna med DIC/WAIC i kombination med annan information om modellernas skattningar.
- Genomsnitt över modeller: använd DIC/WAIC för att konstruera en posterior prediktiv fördelning som tar hänsyn till den relativa prediktiva förmågan bland modellerna.

Modelljämförelse

- Exakt samma observationer måste användas för att jämföra modeller. R kan t.ex. ta bort observationer med Missing values för en modell, men inte för en annan.

- Jämför modellernas DIC/WAIC värden och hur osäkerheten i parametrarna förändras mellan modellerna.

- Akaike-vikter för att jämföra modellers relativa prediktionsförmåga med avseende på WAIC:

$$w_i = \frac{\exp\left(-\frac{1}{2}dWAIC_i\right)}{\sum_{j=1}^{m}\exp\left(-\frac{1}{2}dWAIC_j\right)},$$

där $dWAIC_i = WAIC_i - \min\left(WAIC_k\right)$, $k = 1, \ldots, m$, är skillnaden mellan $WAIC$ för varje modell i och $WAIC$ för den modell med lägst värde på $WAIC$.

Modelljämförelse

- Akaikes tolkning av vikt w_i: skattning av sannolikheten att modell i kommer ge dom bästa prediktionerna på nya data, givet alla modeller som betraktas för Akaike-vikter.

- Akaikes vikter ska ej övertolkas med avseende på t.ex. urvalsstorlek och tidigare resultat om det som undersöks.

- Akaikes vikter för modelljämförelse kan kompletteras med att jämföra parametervärden mellan konkurrerande modeller, vilket t.ex. kan ge svar på

  - varför en modell har lägre WAIC än en annan
  - hur mycket parametervärdena förändras mellan modellerna (confounding effekt?)

Exempel: Modelljämförelse #R kod 3.7

- Linjär regressionsanalys med den beroende variabeln $y =$ liter per mil för en bils bränsleförbrukning, $x_1 =$ hästkrafter i hundratals och $x_2 =$ antal sek på en kvarts mile.

- Jämför 4 linjära regressionsmodeller med följande alternativ på förklaringsvariabler: $x =$ inga, x_1, x_2, (x_1, x_2):

$$Y_1, ..., Y_n | \mu_i, \sigma^2 \overset{iid}{\sim} N(\mu_i, \sigma^2)$$
$$\mu_i = \beta x,$$

- Oberoende priors för $(\beta_0, \beta_j, \ln \sigma)$:

$$\beta_0 \sim N(1, 1)$$
$$\beta_j \sim N(0, 5)$$
$$\ln \sigma \sim N(0, 2),$$

där $j = 1, 2$.

Genomsnitt över modeller

- Prediktioner kan baseras på modellers relativa Akaike-vikter.
- Genomsnittliga posteriorprediktioner över modellers relativa prediktionsförmåga.
- Arbetsgång:
  - Beräkna DIC/WAIC och Akaikes vikt för respektive modell.
  - Beräkna simulerade utfall från respektive modell.
  - Kombinera dom simulerade utfallen från respektive modell med respektive modellvikt som proportion för varje utfall.

Markov chain Monte Carlo (MCMC)

- Skattning av posteriorfördelningar genom användning av en stokastisk process som kallas **Markov chain Monte Carlo (MCMC)**.
- Samplade värden från parametrarnas posteriorfördelning utan krav på att posteriorfördelningen är multivariat normalfördelning och utan att behöva maximera posteriorn.
- **MCMC algoritm** kan skatta generaliserade modeller (t.ex. logitmodell, Poissonregression) och multilevelmodeller som producerar icke-normalfördelade posteriorfördelningar. Kvadratisk approximation är därför olämplig för dessa modeller.
- Extra kostnader med MCMC:
  - tar ofta längre tid att skatta modellen
  - lite mer jobb krävs för att specificera modellen och kontrollera att den skattade modellen ger rimliga resultat.
- **MCMC skattning** av modellen kan göras med hjälp av det probabilistiska programmeringsspråket **STAN**, som **rethinking** paketet kallar på med hjälp av funktionen **map2stan**.

Metropolis algoritmen

- Metropolis algoritmen är ett exempel på en MCMC algoritm. Används för att dra parametervärden från komplicerade posteriorfördelningar.
- Metropolis algoritmen kräver en symmetrisk förslagsfördelning, t.ex. en normalfördelning eller en t-fördelning.
- Exempel: antag att vi vill utvärdera posteriorfördelningen $p(\theta|y)$ för en parameter θ med hjälp av en normalfördelning med varians 1 som förslagsfördelning. För varje iteration t i Metropolisalgoritmen görs följande steg:
  1. ett förslag på θ dras från förslagsfördelningen, givet den förra accepterade dragningen θ^{t-1}. Kalla förslaget $\theta^{\star}$.
  2. Beräkna kvoten r av posteriorfördelningarna för $\theta^{\star}$och θ^{t-1}, dvs
  $$r = \frac{p(\theta^{\star}|y)}{p(\theta^{t-1}|y)},$$
  3. Förslaget accepteras med sannolikheten $\min(r, 1)$. Om förslaget inte accepteras sätts $\theta^{t} = \theta^{t-1}$.

Metropolis-Hastings och Gibbs sampling

- Metropolis-Hastings algoritmen generaliserar Metropolis algoritmen för att tillåta assymetriska förslagsfördelningar.
- Assymetriska förslagsfördelningar kan vara lämpliga för t.ex. variansparametrar som är positiva.
- Gibbs sampling är ofta en variant av Metropolis-Hastings algoritmen och samplar i varje iteration värden på en delmängd av parametrarna, givet dom senaste samplade värdena för dom övriga parametrarna (dvs värden samplas från betingade posteriorfördelningar).
- Syftet med Gibbs sampling är att få effektiva dragningar genom att på ett lättare sätt sampla dragningar från betingade posteriorfördelningar.
- Programmet BUGS (Bayesian Inference Using Gibbs Sampling) använder uteslutande Gibbs sampling för att skatta Bayesianska modeller.
- Gibbs sampling blir dock ofta ineffektivt för modeller med många parametrar och/eller betingade posteriorfördelningar som inte blir lättare att sampla i från.

Hamiltonian Monte Carlo (HMC)

- Mer datorintensiv algoritm än Metropolis-Hastings algoritmen, men dragningar från posteriorfördelningen är ofta mera effektiva.
- Algoritmen är inte en slumpmässig process som för Metropolis-Hastings algoritmen.
- **HMC** kan liknas med att en partikel sveper över hela posteriorn och varje gång partikeln vänder riktning samplas parametervärden från posteriorn vid den aktuella positionen.
- Partikeln färdas långsammare ju mer sannolikhetsmassa posteriorn har. Detta ger en högre andel dragningar där posteriorn har mycket sannolikhetsmassa jämfört med där posteriorn är flack och har liten sannolikhetsmassa.
- **Rethinking** paketets funktion **map2stan** gör användning av **HMC** enkel.

Bearbetning av data och MCMC diagnostik

- Preparera data innan användning av **map2stan**:
  1. alla variabeltransformationer för data ska göras innan. Använd endast de variabler som ska in i modellen.
  2. skapa en ny data frame med alla (transformerade) variabler som ska användas i modellen.

- Hur många dragningar från posteriorn är tillräckligt? Använd MCMC diagnostik:
  - n_{eff} är antalet effektiva dragningar, vilket motsvarar ungefär antalet oberoende dragningar från posteriorn. $n_{eff} > 100$ innebär ofta bra konvergens till posteriorn.
  - *Rhat* är ett mått på om MCMC algoritmen konvergerat eller ej till posteriorn och $Rhat < 1.1$ innebär ofta ok konvergens.

- Använd olika startvärden (olika **MCMC kedjor**) för MCMC algoritmen för att undersöka om respektive kedja utvärderar posteriorn på samma sätt. MCMC kedjorna kan parallelliseras.

Multipel linjär regression - exempel med MCMC #R kod 3.8

- Multipel linjär regressionsanalys med
  - beroende variabel y = liter per mil för en bils bränsleförbrukning
  - x_1 = manuell växellåda (=1)
  - x_2 = vikt i ton
  - x_3 = antal hästkrafter
  - x_4 = tid i sek på en kvarts mile
  - x_5 = antal framåtväxlar
- Standardisera alla förklaringsvariabler förutom dummyvariabeln x_1.

MCMC diagnostik

- **Traceplott** över parameterdragningarna för varje MCMC kedja. Undersök om respektive kedja utvärderar posteriorn på samma sätt. Konventionellt med 3 eller 4 MCMC kedjor till att börja med.
- En plott för det ackumulerade posterior medelvärdet över MCMC samples kan skapas för att kontrollera att det ackumulerade posterior medelvärdet konvergerar till ett värde över dragningarna.
- HMC algoritmen använder en uppvärmningsfas och samples från denna fas kallas *adaptation samples*. Efter denna fas sparas de samplade parametervärdena.
- Jämför resultat för olika mycket *adaptation samples* och olika mycket antal sparade samplade parameterdragningar. Om resultat för fler MCMC dragningar knappt gör någon skillnad, så kan man nöja sig med färre MCMC dragningar.
- Problem med MCMC sampling från posteriorn: pröva med **svagt informativa priors** för att erhålla mer rimliga MCMC kedjor.