732G57 Maskininlärning för statistiker

Föreläsning 11

Josef Wilzén
IDA, Linköping University, Sweden

Dagens föreläsning

- Info
- Del 1:
  - Datahantering
  - Modellering, modellval, hyperparametrar
- Del 2:
  - Lite från gamla föreläsningar
  - Utblickar - tillämpningar och utökningar inom ML
  - Sammanfattning av kursen

Projekt och resten av kursen

- Kursvecka 7 och framåt:
  - Arbeta med projektet
  - Datorlabbar: hjälp med projektet $\rightarrow$ utnyttja tiden!
  - Förbereda inför tentan
- Datum: länk

Välja kurser till vårterminen

Till VT så ska ni välja kurser på 15 HP.
Några rekommendationer om man är intresserad av maskininlärning.

- Om man vill arbeta direkt efter examen:
  - Läs en kurs i programmering i Python
- Om man är intresserad av master i Statistics and Machine Learning
  - 764G03 Flervariabelanalys
  - Någon kurs i optimering
  - Python/Fördjupande kurs i programmering/algoritmer

Del 1:
Datahantering och Modellering

Förstå och förbered data

- Identifiera förklarande variabler och responsvariabel.
- Beskriv vad som utgör en observation i datasetet.
- Hantera kategoriska variabler – slå ihop obalanserade kategorier vid behov.
- Undersök extremvärden – uteslut endast med tydliga regler.
- Hantera saknade värden:
  - Uteslutning är ofta enklast.
  - Imputering kan användas, men med försiktighet.

Dela upp data

- Dela upp i:
  - Träningsdata
  - Valideringsdata
  - Testdata
- Vid klassificering: kontrollera klassfördelning i alla dataset.
- Testdata används först i slutet för att skatta framtida testfel (generaliserbarhet).

Standardisering och transformationer

- Standardisera kontinuerliga variabler vid behov.
- Spara `x_mean_train` och `x_sd_train` för varje kontinuerlig variabel.
- Använd dessa värden för att standardisera validerings- och testdata.
- De flesta problem kräver att vi gör en viss mängd manuella transformationer av vissa variabler (feature engingering).
- Ibland behöver vi transformera responsvariabeln, ex: $y_{log} = log(y)$
- Transformationer bör baseras på träningsdata:
  - Ex: log-transformera variabler.
  - Ex: skapa kategorier utifrån medianvärde i träningsdata.

Praktiskt arbetssätt

- Börja med en kort explorativ fas för att förstå data och möjliga metoder.
- Formulera ett strukturerat upplägg:
  - Hur ska data delas upp?
  - Vilka modeller ska användas?
  - Vilka hyperparametrar ska testas?
- Beskriv detta upplägg tydligt i rapportens metoddel, under Praktisk Metod.

Hyperparametrar – val och optimering

- Vissa hyperparametrar fixeras (tidsbegränsningar, avgränsning).
- Andra optimeras med hjälp av valideringsdata eller korsvalidering.
- Viktiga hyperparametrar bör identifieras via litteratur.
- Ange tydligt:
  - Vad som fixeras
  - Vad som optimeras
  - Vilka värden som testas
  - Vilka mått som används för utvärdering

Loopar för hyperparametersökning

- Skatta flera modeller genom att loopa över vektor/lista med hyperparametervärden
- Spara resultat för träning och validering i vektorer/matriser.
- Jämför modeller baserat på utvärderingsmått.
- Exempel:
  - Testa olika värden på `k` i KNN.
  - Välj det `k` som ger lägst genomsnittligt MSE på valideringsdata.

Utvärdering av modeller

- Hur ska modellerna utvärderas och jämföras?
- Välj lämpliga utvärderingsmått:
  - Klassificering: övergripande och klassvisa mått
  - Regression: övergripande mått och residualanalys
- Visualisera resultat med plottar och tabeller.

Val av utvärderingsmått

- **Klassificering:**
  - Övergripande mått: Träffsäkerhet, felkvot
  - Klassvisa mått: Sensitivitet, Specificitet, Precision, F1-score
  - Viktigt vid **obalanserade** klasser
- **Regression:**
  - Övergripande mått: MSE, MAE
  - Residualanalys: undersök systematiska fel
  - MAE kan vara bättre än MSE vid extrema värden i residualerna

Klassobalans i klassificering

- Vid tydlig obalans i responsvariabeln kan följande metoder användas:
  - **Undersampling:** Minska antalet observationer i majoritetsklassen.
  - **Oversampling:** Öka antalet observationer i minoritetsklassen, t.ex. genom duplicering eller syntetiska exempel (SMOTE).
  - **Viktade kostnadsfunktioner:** Ge högre vikt till minoritetsklassen vid träning.
    - Exempel: viktad log-loss, viktad cross-entropy
- Syftet är att förbättra modellens förmåga att identifiera minoritetsklassen.
- Se kap 6.11 i **IDM**

Användning av valideringsdata

- Valideringsdata används för att välja hyperparametrar.
- Oftast väljer vi den modell som ger lägst fel på valideringsdata givet valda utvärderingsmått
- När bästa kombinationen hyperparametrar hittats för en modell:
  - Skatta om modellen med träningsdata **och** valideringsdata.
  - Antagande: bra hyperparametrar $\rightarrow$ rimlig regularisering.
  - Mer data kan användas utan att riskera överanpassning.

Nästlade valideringsscheman

- Ibland skapas en extra valideringsmängd inom träningsdata.
- Används för att välja hyperparametrar för en specifik modellklass.
- Kräver att man har tillräckligt mycket data.
- Exempel: intern validering i Keras vid träning av neurala nätverk.

Exempel – lasso vs neurala nätverk

- Dela upp data i träning, validering och test.
- **Lasso regression:**
  - Använd korsvalidering på träningsdata för att välja λ.
- **Neurala nätverk:**
  - Skapa intern valideringsmängd inom träningsdata.
  - Välj hyperparametrar för nätverket, exempel:
    - antal gömda lager, antal noder i lager
    - learning rate, batch size
- Jämför modeller på valideringsdata.
- Välj bästa modell och utvärdera på testdata.

Projektet - resultat

- Använd tydliga tabeller för att visa mått för olika modeller/hyperparametervärden
- Presentera resultat för både träningsdata och valideringsdata (i samma tabell)
- Plottar kan exempelvis användas för att illustrera:
  - Modellens prestanda över skattningsiterationer
  - Effekten av olika förklarande variabler
  - Residualer (vid regression)

Revidering av upplägg

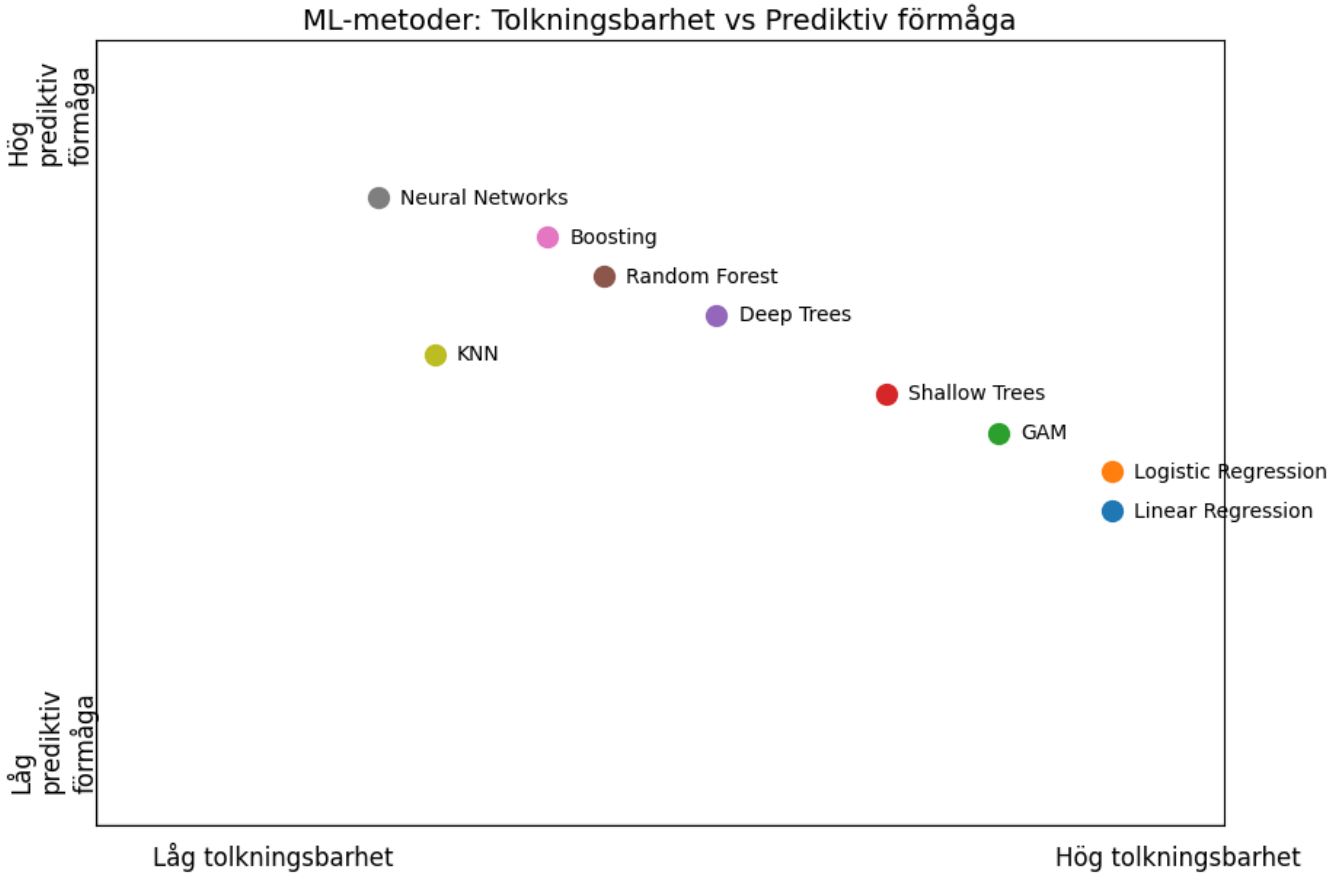
- Det är tillåtet att revidera sitt praktiska upplägg under analysens gång.
- Viktigt att uppdatera beskrivningen av upplägget i rapporten.
- Exempel på revidering:
  - Ändrad datadelning
  - Ny modellklass
  - Justerade hyperparametrar

Tolkningsbarhet vs prediktiv förmåga

- Modeller skiljer sig i hur lätta de är att tolka och hur bra de är på prediktioner.
- **Mer tolkningsbara:**
  - Linjär regression, logistisk regression
  - Grunda trädmodeller, Generalized Additive Models (GAM)
- **Mellanläge:**
  - Djupare träd, Random Forest, Boosting
- **Mindre tolkningsbara:**
  - Neurala nätverk, KNN
- Valet beror på syftet: förklaring eller prediktion?

Tolkningsbarhet vs Prediktiv förmåga

Konceptuell bild över olika modellers egenskaper.



Del 2:
Sammanfattning av kursen, utblickar mm

Rester från tidigare föreläsningar

- Autoencoders
- XGBoost

Utblickar inom maskininlärning

- Kursen fokuserar på grundläggande ML-metoder
- Det finns många avancerade och specialiserade metoder
- Här följer några exempel på utökningar och utblickar

Probabilistisk maskininlärning

- Modellera osäkerhet och variation direkt i sannolikhetsmodellen
- **Normal likelihood**: modellera både medelvärde och varians med flexibla funktioner
- **Poisson likelihood**: för räknevariabler, t.ex. antal händelser
- **Gamma / log-normal**: för positiva och skeva utfall
- **Quantile regression**: modellera olika kvantiler istället för bara medelvärdet

Probabilistisk ML – Flexibel modellering

- Modellera **alla parametrar** i likelihoodfunktionen med flexibla funktioner
- Går bortom antagandet om konstant varians eller fixerad form
- **Normalfördelning**: modellera både medelvärde och varians som funktioner av indata
- **Poissonfördelning**: modellera intensitet (lambda) för räknevariabler
- **Gammafördelning**: modellera både form och skala för positiva, skeva data
- **Log-normalfördelning**: för multiplicativa effekter och skevhet
- **Quantile regression**: modellera olika kvantiler direkt, utan antagande om fördelning

Normal likelihood med neurala nätverk

Antag att vi har n observationer (x_i, y_i) för $i = 1, \ldots, n$. Vi modellerar:

$$y_i \mid x_i \sim \mathcal{N}(\mu(x_i), \sigma^2(x_i))$$

där:

$$\mu(x_i) = f_\mu(x_i; \theta_\mu), \quad \log \sigma^2(x_i) = f_\sigma(x_i; \theta_\sigma)$$

Den totala log-likelihood ges av:

$$\log p(y_1, \ldots, y_n \mid x_1, \ldots, x_n) = -\frac{1}{2} \sum_{i=1}^{n} \left[\log(2\pi\sigma^2(x_i)) + \frac{(y_i - \mu(x_i))^2}{\sigma^2(x_i)} \right]$$

- Två separata nätverk för $\mu(x)$ och $\sigma^2(x)$
- Träning sker genom att maximera log-likelihood över hela datasetet

Bayesianska metoder

- Modellering med sannolikhetsfördelningar
- Ger osäkerhetsmått och möjliggör inferens
- Baseras på Bayes sats: $p(\theta|y, X) \propto p(y|\theta, X) \cdot p(\theta)$
- Generellt ramverk för inferens som kan användas för traditionella metoder och inom ML
- Exempel: Bayesiansk regression, Gaussian Processes regression, BART, Bayesian Deep Learning
- Kräver ofta MCMC eller variational inference
- Se kursen 732G43 Bayesiansk statistik

Tidserieprognoser med maskininlärning

- ML-modeller kan användas för att förutsäga framtida värden i en tidsserie
- Kräver ofta att data omvandlas till ett övervakat format:

$$\text{Input: } (y_{t-1}, y_{t-2}, \ldots, x_{t-1}, x_{t-2}, \ldots), \quad \text{Output: } y_t$$

- Exempel på ML-metoder:
  - Trädmodeller (t.ex. XGBoost)
  - Neurala nätverk (t.ex. MLP, LSTM)
  - Hybridmodeller: ML + ARIMA
- Fördelar: kan hantera icke-linjära samband och flera indata
- Utmaningar: sekventiellt beroende, överanpassning, databehandling

Tolkningsbar maskininlärning

- Målet: förstå hur och varför en modell gör sina prediktioner
- Viktigt inom känsliga tillämpningar: medicin, juridik, finans
- Exempel på metoder:
  - Feature importance (t.ex. permutation, SHAP)
  - Partial dependence plots (PDP)
  - Surrogatmodeller (t.ex. träd som approximerar komplexa modeller)
  - Lokala förklaringar (t.ex. LIME)
- Rekommenderad läsning: Interpretable Machine Learning av Christoph Molnar

Generativa modeller för komplex data

- Generativa modeller skapar ny data baserat på inlärda mönster
- Används för att generera text, bilder, ljud, kod m.m.
- **Transformer-arkitekturen** är central:
  - Självuppmärksamhet (self-attention) för att modellera beroenden
  - Skalbar och effektiv för sekventiell data
  - Grunden för modeller som GPT, BERT, DALL·E, Stable Diffusion
- Textgenerering: GPT, T5, LLaMA
- Bildgenerering: DALL·E, Stable Diffusion, Imagen
- Tränas ofta med stora mängder data och kraftfulla GPU:er

Dimensionreduktion och representation learning

- Syfte: hitta kompakta och informativa representationer av data
- **Dimensionsreduktion**:
  - **t-SNE**: bevarar lokala strukturer, bra för visualisering
  - **UMAP**: bevarar både lokal och global struktur, snabbare än t-SNE
- **Representation learning**:
  - **Autoencoders**: lär latenta representationer genom rekonstruktion
  - **Contrastive learning** (t.ex. SimCLR, CLIP): lär representationer genom att jämföra liknande och olika exempel
  - **Transformerbaserade modeller** (t.ex. BERT, ViT): representationer från sekventiell eller bilddata
- Används för visualisering, klustring, transfer learning och förbehandling

Sammanfattning av kursen

Sammanfattning i en mening:

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

Till vår hjälp har vi gått igenom ett stort antal modeller och algoritmer.

Sammanfattning

- Modellval
  - Felfunktioner
  - Utvärderingsmått
  - Dela upp data i träning, validering, test.
  - Korsvalidering
  - AIC, BIC...
  - Variabelselektion
- Regularisering
  - LASSO, Ridge
  - Vi vill ofta ha så enkla modeller som möjligt
- Vi vill ha bra generaliserbarhet!

Sammanfattning

- Icke-linjär regression/klassificering
  - Grundidé är att hitta en transformationer av förklarande variabler.
  - Gått igenom många olika transformationer.
- Basfunktioner
- Splines
- Kernelfunktioner
- Lokal regression
- Trädmodeller
- Neurala nätverk
  - Olika typer av lager för olika problem
  - Olika aktiveringsfunktioner
  - Global approximation theorem
  - Bra för bilder, video, text, ...

Sammanfattning

- Trädmodeller
  - Dela upp variabelrummet i rektanglar.
  - Varje rektangel får ett värde.
  - Olika regler för uppdelning beroende på problem.
- Beskärning av träd
  - Förbeskärning
  - Efterbeskärning

Sammanfattning

- Ensamblemetoder
  - 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

Sammanfattning

- Klusteranalys
  - Oövervakad inlärning.
  - K-means klustring
  - K-medoid klustring
  - Hierarkisk klustring
  - DBSCAN

Tack för att ni har lyssnat!

Nu är det bara projektet och tentan kvar.