Stockholms universitet

Prediktion av bokföringskonto med en multinomial logistisk regressionsmodell

Henri Jäderberg

Kandidatuppsats i matematisk statistik
Bachelor Thesis in Mathematical Statistics

Kandidatuppsats 2017:24
Matematisk statistik
Juni 2017

www.math.su.se

Matematisk statistik
Matematiska institutionen
Stockholms universitet
106 91 Stockholm

Stockholms universitet

Matematisk statistik
Stockholms universitet
Kandidatuppsats **2017:24**
http://www.math.su.se/matstat

Prediktion av bokföringskonto med en multinomial logistisk regressionsmodell

Henri Jäderberg*

Juni 2017

Sammanfattning

I detta kandidatarbete konstrueras en modell som predikterar vilket bokföringskonto en kostnad bokförs på. Den modell som används är en *Baseline-Category Model*, som är en generalisering av logistisk regression från två till flera klasser. Vid skattning av parametrarna används regularisering av typen Ridge regression. Modellselektion görs genom korsvalidering där förlustfunktioner jämförs. Den slutgiltiga modellen utvärderas med avseende på hur väl den presterar då den anpassas på helt ny data. Modellen predikterar 57.02% av observationerna rätt. Detta kan jämföras med att alltid gissa på det mest förekommande kontot, vilket uppskattningsvis predikterar 18.41% rätt. Prediktionsförmågan varierar dock beroende på vilket konto som observationen faktiskt antar. Mer data har troligtvis endast potential att öka andelen observationer som modellen predikterar korrekt till en nivå på drygt 65%. För att nå högre behövs nya prediktorer.

*Postadress: Matematisk statistik, Stockholms universitet, 106 91, Sverige.
E-post: h.jaderberg@gmail.com. Handledare: Tom Britton och Benjamin Allévius.

Abstract

In this bachelor thesis, a model for predicting booking accounts for costs is constructed. The type of model is a *Baseline-Category Model*, which is a generalization of logistic regression from two to more classes. When estimating the parameters, Ridge regression as regularization method is used. Model selection is done by cross-validation where a loss function is used as a comparative metric. The final model is, thereafter, evaluated by its performance on previously unseen data. The resulting model has an accuracy of 57.02%. This is to be compared with an estimated accuracy of 18.41% for a model always predicting the most common booking account. The model's performance is found to depend greatly on which account the cost is actually booked in. Fitting the model to even more data is estimated to only have a positive effect on accuracy up to a level a little over 65%. For an even higher accuracy, new predictors are needed.

Uppsatsförfattarens tack

Stort tack går till mina handledare, Tom Britton och Benjamin Allévius, som speciellt i början av uppsatsprocessen har gett ovärderlig vägledning och som genom hela processen gett värdefulla tips.

Vidare vill jag tacka företaget Mr. Shoebox. De på företaget jag vill tacka speciellt är följande: Ken Bertling som kom på idén för uppsatsen och som funnits tillgänglig när jag haft frågor som rör bokföring. Tomas Kircher som hjälpte till att skaffa fram data och leverera den i lämpligt format.

Jag vill även tacka min kurskamrat Pavel Lukashin som jag haft intressanta diskussioner med och som även lämnat användbar feedback i arbetets slutfas.

Sist men inte minst vill jag tacka min flickvän, Patricia, som varit ett stöd genom hela uppsatsen.

Innehållsförteckning

1 Inledning

År 1494 publicerades boken *Summa de Arithmetica, Geometria, Proportioni et Proportionalita* av Luca Pacioli som för första gången beskrev bokföringsmetoden som används idag. Denna metod kallas för dubbel bokföring. Även om effektiviseringar skett genom åren kvarstår den största flaskhalsen för kostnadseffektiv bokföring, nämligen att bokföringen sker manuellt av högutbildad arbetskraft. I och med den explosionsartade utvecklingen i datorkraft och den stora mängd data som potentiellt finns tillgänglig råder för första gången förutsättningarna att genom statistiska metoder försöka hitta ett sätt att automatisera bokföring. Det huvudsakliga syftet med denna uppsats är att vara en del av detta arbete.

I detta arbete konstrueras en modell som predikterar utifrån ett antal prediktorer vilket konto en kostnad bokförs på. Eftersom det finns en stor mängd konton, förenklades först problemet genom att utgå från en minimal kontoplan försedd av företaget Mr. Shoebox. Genom vissa ytterligare förenklingar begränsades antalet konton till 19 stycken. Den slutgiltiga modellen som valdes i samband med modelleringen testades därefter på ny data. Modellen lyckas prediktera 57.02% av denna data rätt. Genom en utvärdering av modellen förväntas anpassning på mer data ge en bättre modell, men endast till en nivå där modellen skulle prediktera rätt på som mest drygt 65% av observationerna. För att nå en ännu högre nivå kommer det enligt analysen krävas nya prediktorer som på bättre eller nya sätt kan skilja ut kontona. Ett sekundärt syfte för modellen är att använda det som bokföringsstöd för vissa observationer. Tanken är att om sannolikheten är över 0.9, skulle modellen kunna vara ett stöd för manuell bokföring. Det visade sig att detta gäller för ungefär 11% av observationerna och bland dessa lyckades modellen prediktera 95.08% av kontona korrekt.

1.1 Metoder för klassificeringsproblem med fler än två klasser

Den sorts problem som vi försöker lösa är ett så kallat klassificeringsproblem. En klass definieras som ett av de olika värdena som responsvariabeln kan anta. I vårt fall representerar klasserna de olika kontona som kan antas. Eftersom antalet konton är fler än två behöver vi använda någon metod för klassificeringsproblem med fler än två klasser. För att lösa klassificeringsproblemet används en generalisering av logistisk regression till flera klasser. Den specifika generaliseringen kallas för *Baseline-Category Model*. Val av prediktorer och interaktioner mellan prediktorer väljs enligt en forward selection-

algoritm. För skattning av parametrar används regularisering. För att välja den optimala styrkan i regulariseringen används korsvalidering för en sekvens av olika regulariseringskonstanter. När den slutgiltiga modellen valts utvärderas denna med avseende på hur väl den presterar på ny data.

Enligt Aly (2005) finns det tre huvudsakliga sätt att kategorisera klassificeringsmodeller för fler än två klasser. Det första sättet är att generalisera en binär klassificeringsmodell till flera klasser. Ett exempel på detta är just den modell som används i detta arbete. Det andra sättet är att reducera klassificeringsproblemet till flera binära klassificeringsproblem. Ett enkelt exempel på detta är den så kallade *One-vs-All*-metoden, där man för varje klass löser det binära klassificeringsproblemet om observationen antar klassen eller inte. Därefter kan man välja den klass som fick högst skattad sannolikhet. Det tredje sättet är att använda sig av hierarkisk klassificering. Hierarkisk klassificering går ut på att modellen stegvis reducerar utfallsrummet för observationen ända tills utfallsrummet endast består av en klass.

På ett eller annat sätt kan alla binära klassificeringsmodeller användas för klassificeringsproblem med flera klasser. Detta betyder att det finns en hel uppsjö av möjliga modeller att välja mellan. Anledningen till att modellen som används i detta arbete utgår från logistisk regression har att göra med att det är en relativt enkel modell som är mycket vanlig inom statistiken.

1.2 Problemformulering

Sammanfattningsvis kan problemformuleringen för detta arbete formuleras med hjälp av följande punkter.

1. Konstruktion av en modell av typen *Baseline-Category Model* som predikterar bokföringskonto.
2. Hur väl presterar modellen generellt?
3. Hur väl presterar modellen för de enskilda kontona?
4. Hur kan modellen förbättras enligt punkterna (2) och (3)?
5. Hur väl presterar modellen när det kommer till att användas som bokföringsstöd?

1.3 Arbetets disposition

Arbetet börjar med att presentera teorin som används för konstruktion och utvärdering av modellen. Sedan presenteras den data som används. Därefter

följer den statistiska modellering som leder fram till den slutgiltiga modellen. Denna modell utvärderas sedan utifrån punkterna (2) till (5) i problemformuleringen. Till sist avrundas arbetet med en sammanfattande diskussion utifrån problemformuleringen och det slutgiltiga syftet, automatiserad bokföring.

2 Teori

2.1 Multinomial logistisk regression

I detta avsnitt presenteras teori som leder fram till den multinomiala logistiska modell som kommer att användas i detta arbete.

2.1.1 Generaliserad linjär modell

Multinomial regression är en typ av generaliserad linjär modell och därför börjar vi med att förklara vad detta är. En generaliserad linjär modell är en modell som består av tre komponenter (Agresti, 2002, s.116). Den första kallas för den slumpmässiga komponenten och utgörs av responsvariabelns fördelning så när som på en parameter.

Den andra komponenten är den systematiska komponenten som utgörs av de förklarande variablerna uttryckta som en linjär funktion. Om värdena hos prediktorerna representeras av vektorn $x_1, \ldots, x_P$, så utgörs den systematiska komponenten av $\alpha + \beta_1 x_1 + \cdots + \beta_P x_P$, där $\alpha, \beta_1, \ldots, \beta_P$ är konstanter. Den systematiska komponenten kommer framöver att skrivas på det mer komprimerade sättet $\alpha + \boldsymbol{\beta}^T \boldsymbol{x}$, där $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_P)^T$ och $\boldsymbol{x} = (x_1, \ldots, x_P)^T$

Den tredje komponenten utgörs av länken mellan den systematiska komponenten och väntevärdet hos den slumpmässiga komponenten. Det är alltså länken som bestämmer vilken sorts transformation av väntevärdet för den slumpmässiga komponenten som ska vara en linjär funktion av de förklarande variablerna. Om vi låter funktionen g vara länken och μ vara väntevärdet hos den slumpmässiga komponenten, så kan den generaliserade modellen skrivas som

$$g(\mu) = \alpha + \boldsymbol{\beta}^T \boldsymbol{x}.$$

2.1.2 Multipel logistisk regression

En multipel logistisk regressionsmodell är en generaliserad linjär modell som givet en mängd förklarande variabler, används för att bestämma sannolikheten för en enskild händelse. Genom att låta responsvariabeln anta värdet

1 då händelsen inträffar och värdet 0 då händelsen inte inträffar, gäller det att sannolikheten för händelsen är densamma som väntevärdet för responsvariabeln. Därmed kan vi uttrycka modellen i termer av sannolikheten för händelsen istället för väntevärdet. Benämningen multipel syftar på att vi har fler än en förklarande variabel. Länken man använder mellan den systematiska komponenten och väntevärdet hos den slumpmässiga komponenten är log-odds-funktionen. Om vi låter $p(\boldsymbol{x})$ vara sannolikheten för händelsen givet bestämda värden hos de förklarande variablerna som ges av vektorn $\boldsymbol{x}$, så formuleras modellen av (Agresti, 2002 s.182)

$$\log \frac{p(\boldsymbol{x})}{1 - p(\boldsymbol{x})} = \alpha + \boldsymbol{\beta}^T \boldsymbol{x}. \tag{1}$$

Ett uttryck för $p(\boldsymbol{x})$ kan direkt lösas från (1). Detta uttryck ges av formeln

$$p(\boldsymbol{x}) = \frac{\exp(\alpha + \boldsymbol{\beta}^T \boldsymbol{x})}{1 + \exp(\alpha + \boldsymbol{\beta}^T \boldsymbol{x})}.$$

2.1.3 Multinomial logistisk regression

Det finns flera olika sätt att generalisera logistisk regression till fall där responsvariabeln kan anta fler än två olika värden. Det sätt som används i denna uppsats är en modell som på engelska benämns *Baseline-Category Model* (Agresti, 2002 s.267). Om vi låter nivåerna hos responsvariabeln vara $1, 2, \ldots, K$ så formuleras modellen som

$$\log \frac{p_k(\boldsymbol{x})}{p_K(\boldsymbol{x})} = \alpha_k + \boldsymbol{\beta}_k^T \boldsymbol{x}, \quad k = 1, 2, \ldots K - 1, \tag{2}$$

där basfallet är utfallet K.

Generaliseringen består i att (1) kan ses som ett fall av Baseline-Category Model, nämligen fallet då $K = 2$. Detta inses genom att först skriva $p(\boldsymbol{x})$ som $p_1(\boldsymbol{x})$ för att förtydliga att sannolikheten som skattas i den logistiska regressionen är sannolikheten för utfall 1. Vi låter sedan $p_0(\boldsymbol{x})$ stå för sannolikheten att responsvariabeln antar utfallet 0. Eftersom vi har två disjunkta händelser som tillsammans utgör hela utfallsrummet gäller det att $1 - p_1(\boldsymbol{x}) = p_0(\boldsymbol{x})$ och därmed (1) skrivas om till

$$\log \frac{p_1(\boldsymbol{x})}{p_0(\boldsymbol{x})} = \alpha + \boldsymbol{\beta}^T \boldsymbol{x}.$$

Vi ser tydligt att den logistiska regressionsmodellen är ett fall av Baseline-Category Model där basfallet är utfallet 0.

Till skillnad från den logistiska regressionen krävs det något mer omfattande härledning för att hitta explicita uttryck för sannolikheterna $p_k(\boldsymbol{x})$ i termer av de systematiska komponenterna $\alpha_k + \boldsymbol{\beta}_k^T \boldsymbol{x}$. Vi börjar med att lösa ut $p_k(\boldsymbol{x})$ ur (2) då $k \neq K$. Detta ger oss

$$p_k(\boldsymbol{x}) = \exp(\alpha_k + \boldsymbol{\beta}_k^T \boldsymbol{x}) p_K(\boldsymbol{x}). \tag{3}$$

Eftersom utfallen är disjunkta och utgör hela utfallsrummet, så gäller det att $\sum_{k=1}^{K} p_k(\boldsymbol{x}) = 1$.

$$\begin{aligned}
1 &= \sum_{k=1}^{K} p_k(\boldsymbol{x}) \\
&= \sum_{k=1}^{K-1} \exp(\alpha_k + \boldsymbol{\beta}_k^T \boldsymbol{x}) p_K(\boldsymbol{x}) + p_K(\boldsymbol{x}) \\
&= p_K(\boldsymbol{x}) \left(1 + \sum_{k=1}^{K-1} \exp(\alpha_k + \boldsymbol{\beta}_k^T \boldsymbol{x}) \right) \\
\Leftrightarrow p_K(\boldsymbol{x}) &= \frac{1}{1 + \sum_{k=1}^{K-1} \exp(\alpha_k + \boldsymbol{\beta}_k^T \boldsymbol{x})}
\end{aligned}$$

Genom att använda det uttryck vi fått för $p_K(\boldsymbol{x})$ samt (3) får vi att sannolikheterna för de olika utfallen givet vektorn $\boldsymbol{x}$ ges av

$$p_k(\boldsymbol{x}) = \begin{cases} \frac{\exp(\alpha_k + \boldsymbol{\beta}_k^T \boldsymbol{x})}{1 + \sum_{k=1}^{K-1} \exp(\alpha_k + \boldsymbol{\beta}_k^T \boldsymbol{x})}, & \text{om } k = 1, \ldots, K-1 \\ \frac{1}{1 + \sum_{i=k}^{K-1} \exp(\alpha_k + \boldsymbol{\beta}_k^T \boldsymbol{x})}, & \text{om } k = K. \end{cases} \tag{4}$$

För att få ett enda uttryck för $p_k(\boldsymbol{x})$ kan vi definiera parametern α_K som lika med 0 och $\boldsymbol{\beta}_K$ som en nollvektor. Detta betyder att $\exp(\alpha_k + \boldsymbol{\beta}_k^T \boldsymbol{x}) = \exp(0) = 1$, som i sin tur leder till att vi kan skriva om (4) som

$$p_k(\boldsymbol{x}) = \frac{\exp(\alpha_k + \boldsymbol{\beta}_k^T \boldsymbol{x})}{\sum_{i=1}^{K} \exp(\alpha_k + \boldsymbol{\beta}_k^T \boldsymbol{x})} \tag{5}$$

för $k = 1, 2, \ldots K$. Notera att vi i täljaren ersatt termen 1 med $\exp(\alpha_k + \boldsymbol{\beta}_k^T \boldsymbol{x})$. Notera även att parametern α_K och parametervektorn $\boldsymbol{\beta}_K$ inte skattas i modellen och endast definierats för överskådlighet.

2.1.4 Bayesiansk klassificerare

Den multinomiala logistiska regressionsmodellen ger oss den information vi behöver för att kunna göra en prediktion av vilket utfall som responsvariabeln kommer att anta, givet specifika värden hos de förklarande variablerna som ges av vektorn $\boldsymbol{x}$. Det utfall som enligt modellen uppskattas ha högst sannolikhet att inträffa, är det utfall som kommer att predikteras. Mer formellt kan vi skriva detta som att prediktionsfunktionen Pred antar värdet

$$\text{Pred}(\mathbf{x}) = i, \quad \text{om } p_i(\mathbf{x}) > p_j(\mathbf{x}) \text{ för alla } j \neq i. \tag{6}$$

Denna typ av klassificerare kallas för bayesiansk klassificerare (James et al., 2015, s.37).

2.2 Tränings- och testdata

I situationer då man har tillgång till mycket data är det ofta bäst att dela upp data i tränings- och testdata (Hastie et al., 2016, s.219). En vanlig uppdelning, som även används i detta arbete, är att 80% av datamaterialet används som träningsdata och 20% som testdata. Träningsdata är den del av datamaterialet som man använder för att anpassa modellen på. Testdata är den del av data som är helt oberoende av träningsdata och som man använder för att utvärdera sin slutgiltiga modell. Fördelen med denna uppdelning är att man kan jämföra hur väl modellen presterar på tränings- respektive testdata och på så sätt upptäcka problem med modellen.

2.3 Teori för skattning och modellselektion

2.3.1 Förlustfunktion

Förlustfunktion förstås bäst i en situation där vi har en modell och ett observerat värde av responsvariabeln. Givet det observerade värdet, är modellens utfall förknippat med en förlust. Ju bättre korrespondens mellan modellens utfall och det verkliga värdet hos responsvariabeln, ju mindre bör förlusten vara. En förlustfunktion är en funktion som utifrån en modell och flera observerade värden hos responsvariabeln returnerar ett icke-negativt reellt tal (Held and Bové, 2014, s.192). En förlustfunktion kan användas som ett mått på hur väl en modell presterar på en datamängd.

Valet av förlustfunktion är inte givet, men ofta används den negativa log-likelihood-funktionen multiplicerad med någon konstant. (Hastie et al., 2016, s.221). För att vi senare ska kunna jämföra värdet hos förlustfunktionen mellan data där antalet observationer skiljer sig, väljer vi att multiplicera

log-likelihood-funktionen med $-\frac{1}{N}$, där N är antalet observationer. Förlustfunktionen härleds i bilaga A till att vara

$$F(\boldsymbol{\alpha}, \boldsymbol{B}; \boldsymbol{x}, \boldsymbol{y}) = -\frac{1}{N} \sum_{i=1}^{N} \left(\sum_{k=1}^{K} y_{ik}(\alpha_k + \boldsymbol{\beta}_k \boldsymbol{x}_i) - \log \left(\sum_{k=1}^{K} \exp\left(\alpha_k + \boldsymbol{\beta}_k \boldsymbol{x}_i\right) \right) \right). \tag{7}$$

Här låter vi $\boldsymbol{B}$ vara en $P \times (K-1)$ matris där P är antalet parametrar i modellen exklusive interceptet. Denna matris innehar alla de $K-1$ stycken $\boldsymbol{\beta}$-vektorerna. Faktorn y_{ik} antar värdet 1 om responsvariabeln i den i:te observationen antar kategorin k, annars är den lika med 0. Vidare gäller det att $\boldsymbol{\alpha}$ är en vektor med alla intercepten.

2.3.2 Devians och multinomial devians

Devians definieras som (Agresti, 2002 s.119)

$$D = -2(\ell(\hat{\boldsymbol{\alpha}}, \hat{\boldsymbol{B}}; \boldsymbol{X}, \boldsymbol{y}) - \ell(\boldsymbol{y}; \boldsymbol{X}, \boldsymbol{y})) \tag{8}$$

där $\ell(\boldsymbol{y}; \boldsymbol{X}, \boldsymbol{y}))$ är värdet på log-likelihood-funktionen för den mättade modellen, och $\hat{\boldsymbol{\alpha}}$ och $\hat{\boldsymbol{B}}$ är skattningarna i den modell vi anpassat. Den mättade modellen definieras som den modell som har en parameter för varje observation. Den mättade modellen kommer att skatta sannolikheten för det observerade värdet hos responsvariabeln som en enskild observation antar till 1 och sannolikheten för de andra värden som responsvariabeln kan anta till 0. I bilaga A ser vi att log-likelihood-funktionen kan skrivas som

$$\sum_{i=1}^{N} \sum_{k=1}^{K} y_{ik} \log p_k(\boldsymbol{x}_i).$$

Följande gäller för $\ell(\boldsymbol{y}; \boldsymbol{X}, \boldsymbol{y}))$. För de termer där $y_{ik} = 1$ gäller det att faktorn $\log p_k(\boldsymbol{x}_i)$ är lika med $\log(1) = 0$. I de andra termerna har vi att $\log p_k(\boldsymbol{x}_i)$ är lika med $\log(0)$ som är odefinierat. Vi tolkar ändå dessa termer som $y_{ik} \log p_k(\boldsymbol{x}_i) = 0 \cdot \log(0)$ som 0 eftersom faktorn y_{ik} endast infördes för att på ett kompakt sätt skriva vilka termer $\log p_k(\boldsymbol{x}_i)$ som skulle ingå i summan (se bilaga A). Utifrån detta får vi att

$$\ell(\boldsymbol{y}; \boldsymbol{x}, \boldsymbol{y})) = 0.$$

Deviansen kan nu förenklas till uttrycket

$$D = -2\ell(\hat{\boldsymbol{\alpha}}, \hat{\boldsymbol{B}}; \boldsymbol{X}, \boldsymbol{y}).$$

Denna förenkling betyder att vi kan uttrycka förlustfunktionen i termer av deviansen. Detta är avgörande för att vi ska kunna beräkna förlustfunktionen, i och med att mjukvaran som används beräknar ett mått som kallas multinomial devians. Multinomial devians beräknas som kvoten mellan deviansen och antalet observationer som ingår i beräkningen. Formellt kan vi skriva detta som

$$MD = \frac{D}{N} = -\frac{2}{N}\ell(\hat{\boldsymbol{\alpha}}, \hat{\boldsymbol{B}}; \boldsymbol{X}, \boldsymbol{y})$$

Om vi delar den multinomiala deviansen med 2 så får vi förlustfunktionen (7). Värdet för förlustfunktionen kan därför beräknas med följande formel.

$$F(\boldsymbol{\alpha}, \boldsymbol{B}; \boldsymbol{X}, \boldsymbol{y}) = \frac{MD}{2}$$

2.3.3 Avvägningen mellan systematiskt fel och varians

För att förklara koncepten systematiskt fel och varians är det lättast att utgå från att man har ett problem som kan modelleras med linjär regression. (James et al., 2015, s. 37). Ett vanligt sätt att bestämma felet i en modell är att beräkna medelkvadratfelet på oberoende testdata. Medelkvadratfelet definieras som (James et al., 2015, s.29)

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2.$$

Man kan visa att det förväntade medelkvadratfelet kan brytas ner till

$$E[MSE] = Var(\hat{y}_i) + (E[y_i] - E[\hat{y}_i])^2 + Var(\epsilon),$$

där $Var(\hat{y}_i)$ är variansen hos estimatorn för $\hat{y}_i$. Differensen $E[y_i] - E[\hat{y}_i]$ kallas ofta för systematiskt fel. Till sist har vi $Var(\epsilon)$ som är variansen hos feltermen i regressionsmodellen och representerar den naturliga variansen kring det sanna väntevärdet $E[y_i]$. Denna term kan man inte reducera bort.

Det finns alltså två olika sätt att reducera MSE, antingen minskar man variansen hos estimatorn för skattningen av y_i eller minskar man systematiskt fel. Problemet är att dessa två olika sätt ofta är i konflikt. Om vi har relativt många prediktorer i en modell är systematiskt fel ofta låg eftersom den förväntade skattningen av y bör ligga närmare det sanna värdet ju fler prediktorer vi har att utgå ifrån. Däremot kan det vara så att modellen tenderar att anpassa för mycket utefter den data som används vid anpassningen av parametrarna. Detta kan göra så att modellen passar endast för just detta datamaterial, men inte generellt för ett representativt datamaterial. Detta

tar sig i uttryck i att $Var(\hat{y}_i)$ är hög. Intuitivt kan det förstås som att ifall vi skulle använda ny oberoende data att anpassa modellen så är det stor sannolikhet att vi får en skattning för y som skiljer sig relativt mycket från skattningen vi fick tidigare. Om vi har relativt få prediktorer får vi ett omvänt problem. Problemet med hög varians och låg systematiskt fel kallas ofta för överanpassning, och det omvända problemet kallas för underanpassning.

Även om man i klassificering inte använder sig av MSE, gäller motsättningen mellan varians och systematiskt fel (James et al., 2015, s.37).

2.3.4 Korsvalidering

Utöver tränings- och testdata tar man ofta ut en del av observationerna som valideringsdata. Genom att beräkna förlustfunktionen på valideringsdata för modeller man anpassat på träningsdata får vi ett mått som undviker problemet med att avväga systematiskt fel och varians. (Hastie et al., 2016 s.219).

Ett annat sätt att undvika problemet med att avväga systematiskt fel och varians är att använda sig av korsvalidering. Detta går ut på att man delar upp träningsdata i lika stora delar. Standard är att antalet delar är antingen fem eller tio stycken. Vi låter antalet delar vara lika med d. För varje kombination med $d-1$ delar anpassar vi modellen och beräknar sedan förlustfunktionen på den resterande delen. Vi tar sedan genomsnittet av de d stycken reella tal som beräknats (James et al., 2015, s.184).

Fördelen med att använda korsvalidering istället för att endast ha valideringsdata grundar sig på två punkter. För det första använder korsvalidering generellt sätt fler observationer vid varje modellanpassning. Detta betyder att vi får säkrare skattningar av de ingående parametrarna. Den andra fördelen har att göra med att beroende på vilka observationer som blir tränings- respektive valideringsdata, kan värdet på förlustfunktionen variera en hel del. Korsvalidering producerar ett mått som är ett genomsnitt av flera värden beräknade med hjälp av förlustfunktionen, och har därmed en lägre varians. En nackdel med korsvalidering är att det kräver mer uträkningar och därmed mer datorkraft. På grund av detta används fem istället för tio delar, i den korsvalidering som görs (James et al., 2015, s.178).

Man skulle kunna tycka att när man använder korsvalidering, behövs det ingen testdata. Anledningen till att man brukar ha testdata kvar, är att vissa modeller kommer att få lågt förlustfunktionsvärde i korsvalideringen av ren slump. Det finns alltså en risk att vi väljer en modell pga. denna effekt och genom att ha testdata har man ett sätt att upptäcka detta.

2.3.5 Forward selection

Forward selection en mycket vanlig algoritm för modellselektion. Den går ut på att man börjar med en modell utan några förklarande variabler, en så kallad nollmodell. Utifrån denna modell anpassar man en ny modell för varje prediktor, där man lagt till prediktorn till nollmodellen. Utifrån något mått väljer man den modell som presterar bäst av alla dessa modeller. Man fortsätter sedan att utifrån denna modell lägga till ännu en prediktor på liknande sätt. Algoritmen stannar av då den modell man utgår från presterar bättre än alla de utvidgade modellerna (Sundberg, 2016, s.71).

2.3.6 Regularisering

Med regularisering menas olika sätt att förändra skattningen av modellens ingående parametrar så att dessa tenderar att vara närmare noll än de annars hade varit. Genom att använda regularisering kan man undvika överanpassning (James et al., 2015, s.214). Den regulariseringsteknik som används i denna uppsats är Ridge regression. Den går ut på att istället för att maximera log-likelihood-funktionen,

$$\ell(\boldsymbol{\alpha}, \boldsymbol{B}; \boldsymbol{X}, \boldsymbol{y}) = \sum_{i=1}^{N} \left(\sum_{k=1}^{K} y_{ik}(\alpha_k + \boldsymbol{\beta}_k \boldsymbol{x}_i) - \log \left(\sum_{k=1}^{K} \exp\left(\alpha_k + \boldsymbol{\beta}_k \boldsymbol{x}_i\right) \right) \right),$$

maximerar vi istället

$$\begin{aligned} \ell_{RIDGE}(\boldsymbol{\alpha}, \boldsymbol{B}; \boldsymbol{X}, \boldsymbol{y}) = &\sum_{i=1}^{N} \left(\sum_{k=1}^{K} y_{ik}(\alpha_k + \boldsymbol{\beta}_k \boldsymbol{x}_i) - \log \left(\sum_{k=1}^{K} \exp\left(\alpha_k + \boldsymbol{\beta}_k \boldsymbol{x}_i\right) \right) \right) \\ &- \lambda \sum_{k=1}^{K-1} \sum_{p=1}^{P} \beta_{kp}^2. \end{aligned} \tag{9}$$

Här låter vi β_{kp} stå för den p:te parametern i den k:te delmodellen. Konstanten λ kallas för regulariseringskonstant, och beroende på storleken hos denna får vi olika skattningar för modellens parametrar. Hela termen som lagts till log-likelihood-funktionen kallas för regulariseringsterm. Konsekvensen av regulariseringstermen är att om man låter en parameter få större absolutbelopp, kommer detta ge ett lägre funktionsvärde hos ℓ_{RIDGE}. Detta betyder att vid maximering av ℓ_{RIDGE} kommer koefficienterna närmare noll än vid maximering av ℓ. Värt att notera är att storleken hos intercepten inte bestraffas av regulariseringstermen. Det är standardutförande och intuitivt kan

man förstå detta genom att tänka sig att man vill att regulariseringen ska ha en utjämnande effekt på funktionskurvan hos log-likelihood-funktionen. Intercepten påverkar inte formen hos denna kurva utan endast dess läge, och därmed är det ingen idé att bestraffa storleken hos denna.

En annan anledning till att använda regularisering är att vi försäkras om att inga parametrar går mot oändligheten när log-likelihood-funktionen maximeras. Ett exempel på detta är ifall vi har prediktorn x som endast kan anta värdet 0 eller 1. I de träningsdata där x antar värdet 1 har responsvariabeln antagit klassen c. Klass c antas inte vara klassen som utgör basfallet i modellen. Vi utgår nu från följande sätt att uttrycka log-likelihood-funktionen (se bilaga A).

$$\ell(\boldsymbol{\alpha}, \boldsymbol{B}; \boldsymbol{X}, \boldsymbol{y}) = \sum_{i=1}^{N} \sum_{k=1}^{K} y_{ik} \log \left(\frac{\exp\left(\alpha_k + \boldsymbol{\beta}_k^T \boldsymbol{x}_i\right)}{\sum_{i=1}^{K} \exp\left(\alpha_k + \boldsymbol{\beta}_k^T \boldsymbol{x}_i\right)} \right).$$

Uttrycket som logaritmfunktionen appliceras på är inget annat än den skattade sannolikheten för klass k. Vi noterar nu att för de observationer där $x = 0$, påverkas inte uttrycket innanför logaritmfunktionen av storleken på koefficienten för prediktorn x, som vi betecknar som β_x. Detta eftersom $0 \cdot \beta_x = 0$ för alla värden på β_x. Detta betyder att vi endast behöver studera hur β_x påverkar de termer i log-likelihood-funktioner som motsvarar de observationer där $x = 1$. Låt oss säga att antalet sådana observationer är lika med $N_{x=1}$. Delsumman av log-likelihood-funktionen ges nu av

$$\begin{aligned} \ell_{x=1}(\boldsymbol{\alpha}_{x=1}, \boldsymbol{B}_{x=1}; \boldsymbol{X}_{x=1}, \boldsymbol{y}_{x=1}) &= \sum_{i=1}^{N_{x=1}} y_{ic} \log \left(\frac{\exp\left(\alpha_c + \boldsymbol{\beta}_c^T \boldsymbol{x}_i\right)}{\sum_{i=1}^{K} \exp\left(\alpha_k + \boldsymbol{\beta}_k^T \boldsymbol{x}_i\right)} \right) \\ &= \sum_{i=1}^{N_{x=1}} \log \left(\frac{\exp\left(\alpha_c + \boldsymbol{\beta}_c^T \boldsymbol{x}_i\right)}{\sum_{i=1}^{K} \exp\left(\alpha_k + \boldsymbol{\beta}_k^T \boldsymbol{x}_i\right)} \right) \end{aligned} \tag{10}$$

Här låter vi indexeringen $x = 1$ i log-likelihood-funktionens argument notera att vi endast utgår från en delmängd av observationerna. Notera att observationerna som motsvarar delsummorna nu har fått nytt index från 1 till $N_{x=1}$. Notera vidare att vi har tagit bort summeringen över alla K klasser eftersom det enligt antagande endast finns en klass, klass c, där $x = 1$ förekommer. Eftersom $y_{ic} = 1$ för alla delsummor, kan vi ta bort denna faktor.

I bilaga B visas att summanden i (10) är strikt växande med avseende på β_x. Detta betyder att summan är strikt växande med avseende på β_x. Eftersom koefficienten endast har påverkan på denna delsumma av den totala log-likelihood-funktionen, så betyder detta att log-likelihood-funktionen som helhet växer strikt som funktion av β_x. Om vi nu försöker skatta koefficienten

genom att maximera log-likelihood-funktionen så kommer skattningen bli odefinierad, eftersom skattningen blir oändligt stor.

Vi går nu över till fallet då vi skattar parametrar med hjälp av en regulariseringsterm. Termerna som motsvarar observationerna där $x = 1$ utgörs nu av

$$\ell_{x=1}(\boldsymbol{\alpha}_{x=1}, \boldsymbol{B}_{x=1}; \boldsymbol{X}_{x=1}, \boldsymbol{y}_{x=1}) = \sum_{i=1}^{N_{x=1}} \log\left(\frac{\exp\left(\alpha_c + \boldsymbol{\beta}_c^T \boldsymbol{x}_i\right)}{\sum_{i=1}^{K} \exp\left(\alpha_k + \boldsymbol{\beta}_k^T \boldsymbol{x}_i\right)}\right) - \lambda \sum_{p=1}^{P} \beta_{cp}^2$$

Någon av koefficienterna i summan som utgör regulariseringstermen består av β_x. Den partiella derivatan för regulariseringstermen med avseende på β_x är därmed lika med $-2\lambda\beta_x$. Denna derivata går mot minus oändligheten då vi låter β_x gå mot oändligheten. Då den första termens derivata går mot 0 och andra termens derivata går mot minus oändligheten då β_x går mot oändligheten, så kommer funktionen efter ett visst värde hos β_x att övergå från en växande till en avtagande funktion. Detta betyder med andra ord att vi kommer kunna hitta ett reellt tal för β_x när vi försöker maximera denna funktion.

Standardisering av prediktorer

Om man vill att alla prediktorernas parametrar ska straffas likvärdigt behöver man standardisera prediktorerna innan man anpassar modellen (James et al., 2015, s.217). I slutändan har detta att göra med att ju större absolutbelopp en parameter har, desto mer straffas en ökning i absolutbeloppets storlek. Detta betyder att parametrar med stora absolutbelopp straffas hårdare än parametrar med små absolutbelopp. En annan anledning till standardisering är att man får en anpassning till modellen som är oberoende av vilken skala som de ingående prediktorerna har. Vi vill t.ex. att skattningen av parametrarna ska vara densamma även om vi uttrycker variabeln *pris* i ören istället för kronor. Standardiseringen som används är

$$x_{ij}^* = \frac{x_{ij}}{\sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_{ij} - \bar{x}_j)^2}}.$$

Vänsterledet är det standardiserade värdet hos prediktorn j i observation i. Två saker är värda att notera. Det första är att man använder sig av den icke väntevärdesriktiga skattningen av variansen. Den andra är att man i

täljaren inte subtraherar x_{ij} med stickprovsmedelvärdet $\bar{x}_j$, vilket man ofta gör i andra sammanhang.

2.4 Teori för modellutvärdering

2.4.1 Exakthet

Exakthet definieras i denna uppsats som andelen observationer där modellen predikterar rätt (Metz, 1978). På engelska benämns detta mått som *accuracy*. Formellt kan vi skriva detta som

$$Acc = \frac{1}{N}\sum_{i=1}^{N} I(y_i = \text{Pred}(\boldsymbol{x_i})).$$

Funktionen Pred är den Bayesianska klassificeraren som definierats enligt (6). Vidare är I en indikatorvariabel som antar värdet 1 om påståendet $y_i = P(\boldsymbol{x_i})$ är sant. Detta mått är med främst för att svara på den intressanta frågan: Hur ofta förespår modellen rätt? Som mått för hur väl modellen presterar är exakthet begränsat. Ett typiskt exempel på detta är ett klassificeringsproblem med två klasser, varav den ena klassen utgör 99% av datamaterialet. Ifall modellen predikterar denna klass för varje gång, så blir värdet på exakthet 0.99. Vi får alltså ett högt värde för en modell som vi skulle anse inte riktigt tillför något intressant.

Förväntad exakthet definieras som väntevärdet av *Acc*.

2.4.2 Wald-konfidensintervall

Ett Wald-konfidensintervall utgår från Wald-statistikan som definieras som (Held and Bové, 2014 s.99)

$$\sqrt{I(\hat{\theta}_{ML})}(\hat{\theta}_{ML} - \theta) \overset{a}{\sim} N(0,1).$$

Här står funktionen I för fisherinformationen och θ för den parameter som skattas. Värdet $\hat{\theta}_{ML}$ är ML-skattningen av parametern θ. Utifrån statistikan konstrueras Wald-konfidensintervallet till

$$CI_{WALD} = \hat{\theta}_{ML} \pm \frac{\lambda_{0.025}}{\sqrt{I(\hat{\theta}_{ML})}}, \tag{11}$$

där $\lambda_{0.025}$ är normalfördelningens 0.025-kvantil. Ifall parametern som ska skattas är θ i binomialfördelningen, som ges av sannolikhetsfunktionen

$$p(x) = \binom{N}{x}\theta^x(1-\theta)^{N-x} \quad x = 0, 1, \ldots, N,$$

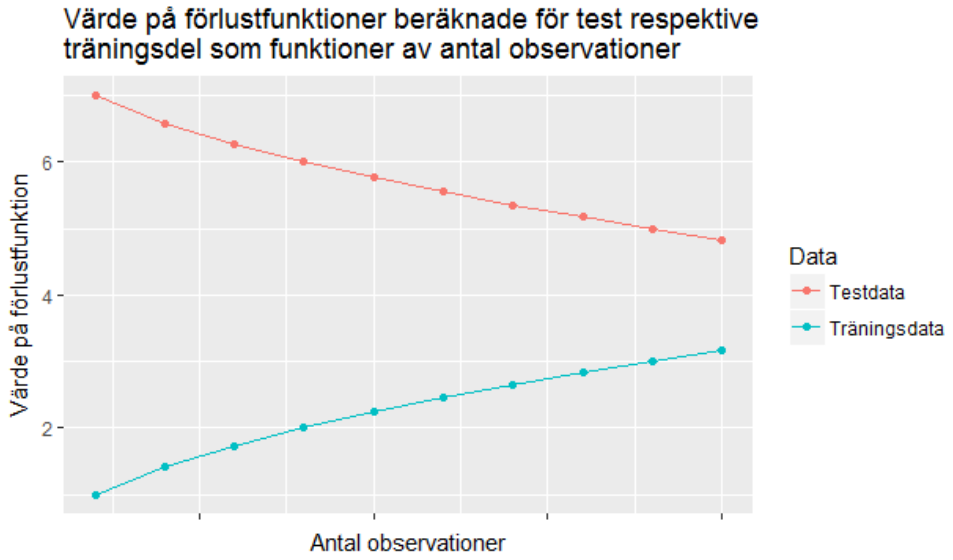
så ges Wald-konfidensintervallet i (11) av

$$CI_{WALD} = \hat{\theta}_{ML} \pm \lambda_{0.025}\sqrt{\frac{1}{N}\hat{\theta}_{ML}(1-\hat{\theta}_{ML})}, \tag{12}$$

där $\hat{\theta}_{ML} = \frac{x}{N}$. Härledningen av detta konfidensintervall finns i bilaga C

2.4.3 Inlärningskurvor

Med inlärningskurvor menas kurvor som visar hur modellen presterar utifrån hur mycket träningsdata som används vid anpassningen (Perlich, 2011). I vårt fall används värdet på förlustfunktionen då den beräknas med hjälp av testdata som mått på hur väl modellen presterar. Om man kompletterar denna kurva med en kurva för värdet av förlustfunktionen då denna istället beräknas på den data man anpassade modellen med, kan man få vägledning ifall mer data eller fler prediktorer skulle kunna förbättra modellen. Figur 1 illustrerar det typiska utseendet för dessa kurvor (Ng).



Figur 1: Det typiska utseendet för förlustfunktionsvärdet beräknad på test och träningsdata som en funktion av antalet observationer som finns i träningsdata

Utseendet hos figuren kan förstås genom följande tre påståenden.

1. Värdet på förlustfunktionen beräknad på testdata går ner med antalet observationer.

2. Värdet på förlustfunktionen beräknad på den data som användes för att anpassa modellen går upp med antalet observationer.

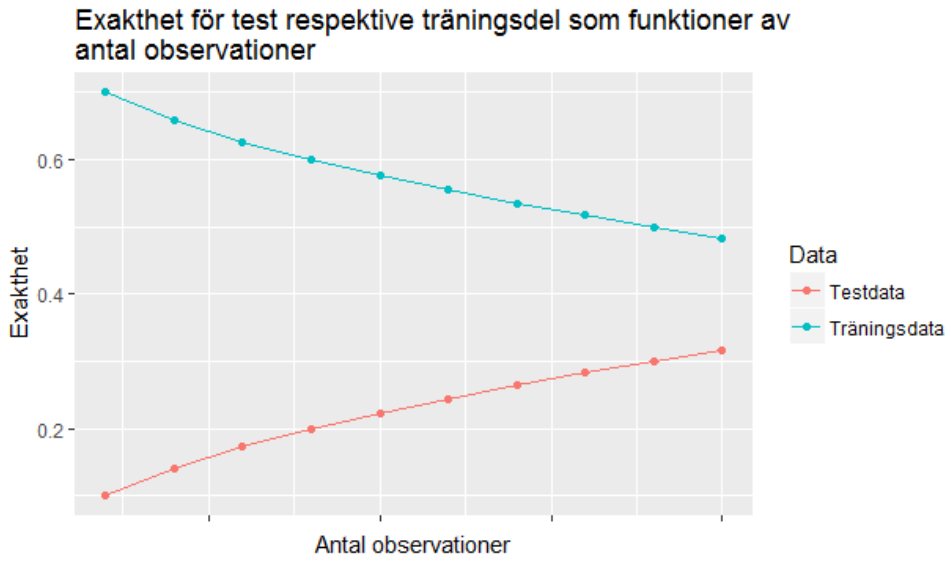
3. Generellt gäller det att förlustfunktionen beräknad på testdata är större eller lika med förlustfunktionen beräknad på data som användes för att anpassa modellen. Detta gäller oberoende av hur många observationer som använts för att anpassa modellen.

Punkt nummer ett kan intuitivt förstås som att ju större mängd data som modellen anpassas på, desto större sannolikhet är det att modellens parametrar är sådana att de överensstämmer med de sanna parametrarna. Detta eftersom variansen hos estimatorerna som ger parameterskattningarna blir mindre. Kopplar vi detta till problemet med avvägning mellan systematiskt fel och varians som presenterades tidigare, så går variansen ner då antalet observationer ökar, medan det systematiska felet är oförändrad. Detta leder till att det förväntade värdet hos förlustfunktionen går ner. Punkt nummer två kan intuitivt förstås som att i och med att antalet observationer ökar, blir det svårare för modellen att anpassa sig så att den stämmer väl överens med alla observationer. Detta gör att det förväntade värdet på den beräknade förlustfunktionen på data som används för anpassning går upp. Punkt nummer tre gäller eftersom modellen bör prestera bättre på data som den anpassades på än helt ny data.

Ifall vi ökar antalet observationer ytterligare kan vi förvänta oss att kurvorna fortsätter i samma riktning som tidigare. Ifall gapet mellan förlustfunktionen för träningsdata och förlustfunktionen för testdata är stor där kurvorna slutar, och lutningen hos förlustfunktionen för testdata är relativt brant, kan vi förvänta oss att fler observationer kommer leda till en bättre modell. Ifall gapet är litet eller lutningen för testdata är relativt flack, kan vi förvänta oss att mer data har en relativt liten inverkan. Om kurvorna ser ut att konvergera mot ett värde på förlustfunktionen som anses för högt när man låter antalet observationer gå mot oändligheten, så räcker det alltså inte att endast öka antalet observationer. Man behöver då även minska det systematiska felet, vilket kan göra med att introducera nya prediktorer som på ett bättre sätt kan skilja ut de olika klasserna i responsvariabeln.

Ett problem med att använda förlustfunktionen som mått på hur väl modellen presterar är att det inte är helt uppenbart vilket värde som anses vara bra. I detta hänseende är exakthet bättre och därför kan det vara rimligt

att komplettera lärningskurvor med exakthet istället. Eftersom ett högt värde på exakthet, till skillnad från för förlustfunktionen, är förknippat med en bättre modell, så får kurvorna formen som ges av Figur 2.



Figur 2: Det typiska utseendet för exakthet beräknad på test och träningsdata som en funktion av antalet observationer som finns i träningsdata

2.4.4 Felmatris

Vi ska här definiera två olika typer av matriser som på engelska kallas för *coincidence matrices*. Den svenska benämningen som används i denna uppsats är felmatris. Vi börjar med felmatrisen för ett klassificeringsproblem med fler än två klasser. Elementet på rad i och kolumn j i denna matris är lika med antalet observationer i testdata som i verkligheten tillhör klass i men som enligt modellen predikterades som klass j (Olson and Delen, 2008, s.31). Applicerat på vårt problem blir denna felmatris en 19×19 matris. Summan av diagonalen i denna matris ger oss antalet observationer som predikterades rätt av modellen.

Den andra felmatrisen vi definierar är en felmatris för ett binärt klassificeringsproblem. Denna felmatris konstrueras med avseende på en viss klass. Om felmatrisen formuleras som

$$C = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \quad ,$$

så motsvarar den följande frekvenstabellen.

	Klassen predikteras	Klassen predikteras inte
Antar klassen	a	b
Antar inte klassen	c	d

2.4.5 Känslighet

Känslighet för klass i definieras som andelen av observationerna som antar klass i som även predikteras att anta klass i (Olson and Delen, 2008, s.138). Formellt sätt är det enklast att utgå från den andra typen av felmatris som definierades i föregående avsnitt. Då definieras känslighet som

$$Se = \frac{a}{a+b}.$$

Dessa mått kan beräknas för varje konto och tillåter oss att se hur väl modellen predikterar när det kommer till olika klasser.

3 Data

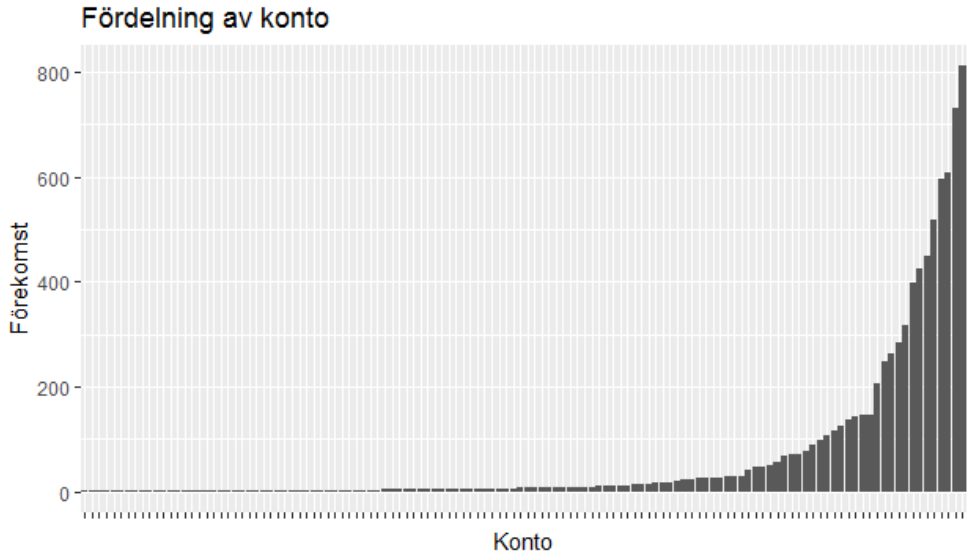
Datamaterialet består av 8154 observationer och 9 variabler då responsvariabeln är inräknad. Detta delas sedan upp i träningsdata som utgör 6530 av observationerna och testdata som utgör 1624 av observationerna. I detta avsnitt presenteras data utan att särskilja mellan tränings- och testdata.

3.1 Beskrivning av responsvariabel

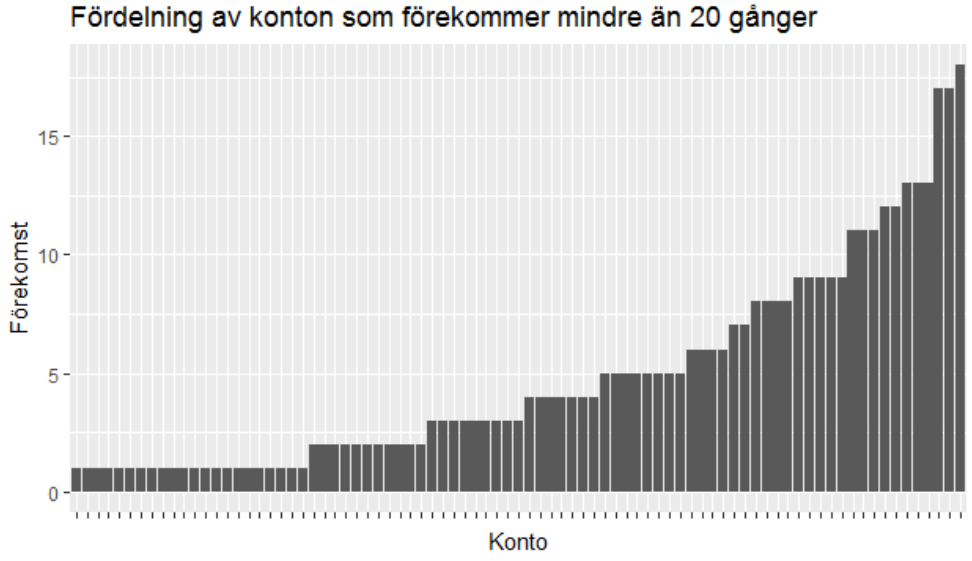
Varje observation utgörs av en bokföringshändelse. Responsvariabeln som används för modellen är det konto som händelsen bokförts på. I datamaterialet finns det totalt 124 konton. Fördelningen för kontona kan ses i Figur 3, då kontona har sorterats från minst förekommande till mest förekommande.

I stapeldiagrammet är det svårt att se frekvensen för de kontona som förekommer väldigt sällan. I Figur 4 plottas endast de kontona som förekom som mest 20 gånger.

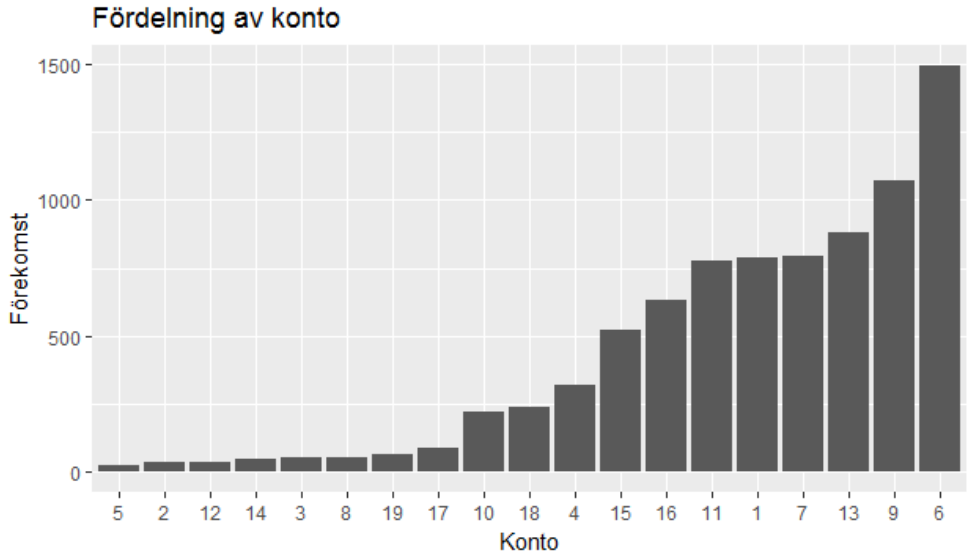
Det finns två egenskaper hos responsvariabeln som är besvärliga för vår analys. Den första egenskapen är att datamaterialet är obalanserat i och med att förekomsten av konton skiljer sig så kraftigt. Den andra egenskapen är att responsvariabeln antar ett stort antal olika konton. Problemen kan beskrivas med begreppet *number of events per variable* som i en binär logistisk regression är lika med antalet ingående variabler i modellen delat med hur många gånger den minst förekommande händelsen förekommer i datamaterialet. Det



Figur 3: Fördelningen hos förekomsten av konton i datamaterialet sorterat från minst förekommande till mest förekommande.



Figur 4: Förekomsten av varje konto bland konton som i datamaterialet förekommer som högst 20 gånger. Kontona är sorterade från minst förekommande till mest förekommande.



Figur 5: Fördelningen hos förekomsten av de 19 slutgiltiga kontona i datamaterialet sorterade från minst förekommande till mest förekommande.

har visat sig att när denna kvot är mindre än 10 uppstår problem såsom att skattningarna av de ingående parametrarna är icke-väntevärdesriktiga samt under- eller överskattat standardfel (Peduzzi et al., 1996). Dessutom tenderar en logistisk regression att underskatta sannolikheten för sällsynta händelser, vilket bokföring på vissa sällsynta konton kan anses vara (King and Zeng, 2001). Dessa problem antas kunna vara problem även i vår multinomiala logistiska regressionsmodell.

För att förenkla problemet slås konton ihop utifrån ett förslag till en minimal kontoplan. Detta resulterar i 31 konton. Tyvärr finns det fortfarande konton som har endast fåtal observationer och den slutgiltiga åtgärden är att slå ihop de kontona som har 20 eller färre observationer. Detta resulterar i en responsvariabel som antar 19 olika konton. Dessa konton namnges som konto 1 till 19. I Figur 5 ser vi förekomsten hos kontona sorterade från minst förekommande till mest förekommande.

Som man ser i Figur 5 är datasetet fortfarande obalanserat, men inte alls till den grad som det var tidigare. En invändning mot att slå ihop konton är att modellen inte nödvändigtvis predikterar det mest sannolika kontot av de ursprungliga 31 i den minimala kontoplanen. Sannolikheten för att en observation ska anta det sammanslagna kontot representerar sannolikheten att observationen antar något av de sammanslagna kontona. Låt oss säga att vi har slagit samman kontona A, B och C. Det kan vara så att modellen

skulle ha predikterat konto D för en observation men nu predikterar det sammanslagna kontot eftersom summan av de skattade sannolikheterna för A, B och C är större än den skattade sannolikheten för D. Detta förväntas dock inte vara ett stort problem i praktiken.

3.2 Beskrivning av prediktorer

I Tabell 1 sammanfattas de åtta prediktorerna som finns med i datamaterialet. Prediktorn *antal artiklar* är strikt sätt inte en kontinuerlig variabel

Prediktor	**Typ**	**Faktornivåer**	**Antal observationer**
pris	Kontinuerlig	-	-
antal_artiklar	Kontinuerlig	-	-
betalningssätt	Faktor	Kontant Kort Okänt/Annat	535 6059 1560
arbetsdag	Faktor	Arbetsdag Inte arbetsdag	5718 2436
köpare_sni	Modifierad faktorvariabel	21 nivåer: Bokstäverna A till U	Se i avsnittet om SNI-prediktorer
säljare_sni	Modifierad faktorvariabel	21 nivåer: Bokstäverna A till U	Se i avsnittet om SNI-prediktorer
momskategori	Faktor	0.25 0.12 0.06 0 Okänd	Se avsnitt om arbetstid och momskategori
arbetstid	Faktor	Arbetstid Inte arbetstid Okänd/Odefinierad	Se avsnitt om arbetstid och momskategori

Tabell 1: Tabell som sammanfattar de åtta prediktorerna som finns tillgängliga i datamaterialet.

eftersom den endast kan anta ett naturligt tal. Dock behandlas den som en kontinuerlig variabel. Betalningssätt avser vilken sorts betalning som gjordes

i samband med kostnaden. I de observationer där betalningssättet antar nivån Okänt/Annat handlar främst om observationer där bokföringsunderlaget varit en faktura.

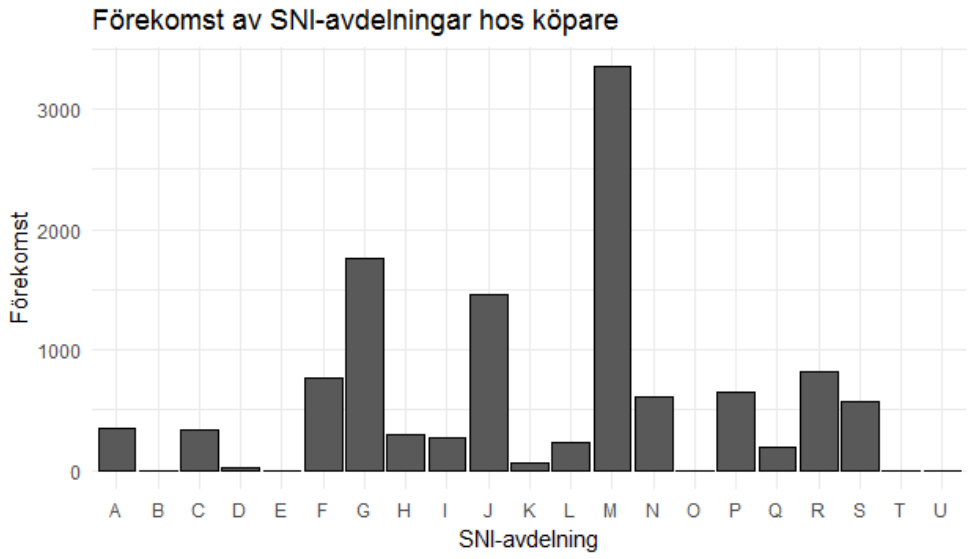
3.2.1 SNI-prediktorer

Prediktorerna *köpare_sni* och *säljare_sni* anger vilken eller vilka SNI-avdelningar som företaget som bokför händelsen har, respektive företaget som köparen interagerat med i händelsen har. Normalt sätt är *säljare_sni* företaget som företaget som bokför händelsen köpt en vara eller tjänst ifrån. Med SNI-avdelning menas den grövsta uppdelning som finns av SNI-koder. SNI-kod är en kod som ett företag har som beskriver företagets verksamhetsområde (SCB (Statistiska Centralbyrån), 2017). Det finns 821 olika SNI-koder vilket för detta arbete bedömdes vara för många för att ha som prediktorer. Lösningen var att istället använda sig av SNI-avdelning. Ett problem med de två SNI-prediktorerna är att ett företag kan ha flera SNI-koder och därför kan man inte använda dessa som normala faktorvariabler. Det man kan göra är att försöka generalisera det sätt man annars hanterar faktorvariabler. När en faktorvariabel anpassas i en logistisk regressionsmodell låter man ena faktornivån fungera som basfall och sedan har man en dummy-variabel för varje av de resterande nivåerna. Om ett företag har s SNI-koder kan det vara rimligt att anta att varje enskild SNI-kod har $\frac{1}{s}$ så stor inverkan som om företaget endast hade denna SNI-kod. Utifrån denna logik så skulle dummy-variabeln för en SNI-avdelning anta kvoten mellan antalet SNI-koder som företaget har i avdelningen och antalet SNI-koder som företaget har överhuvudtaget. För att klargöra detta presenteras några exempel i Tabell 2. Där antas det att det endast finns 3 SNI-avdelningar, A, B och C. SNI-avdelningen A betraktas här som basfallet. De första tre exemplen är fall där företaget endast har en SNI-kod. Detta ger exakt samma resultat som om SNI-prediktorn endast kunde ha en SNI-kod. De två sista exemplen visar vilka värden som dummy-variablerna för B och C antar i två fall där företaget har flera SNI-koder.

SNI-koder i företag	Dummyvariabel för B	Dummyvariabel för C
1 avd. A	0	0
1 avd. B	1	0
1 avd. C	0	1
1 avd. A, 1 avd. B, 1 avd. C	$\frac{1}{3}$	$\frac{1}{3}$
3 avd. A, 1 avd. B, 0 avd. C	0	$\frac{1}{4}$

Tabell 2: Exempel som illustrerar hur SNI-prediktorerna fungerar.

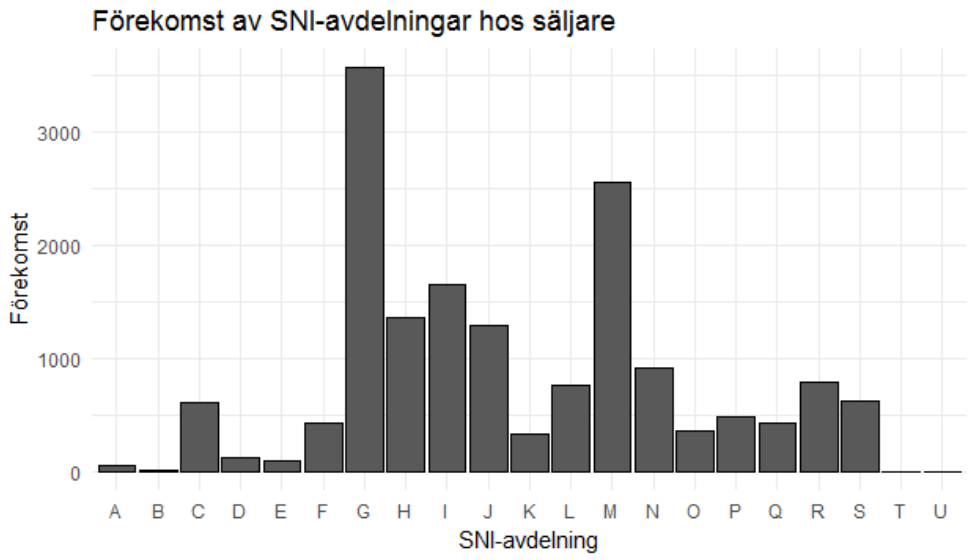
I Figur 6 presenteras förekomsten av köparens SNI-avdelningar bland observationerna i datamaterialet. Eftersom flera SNI-avdelningar kan förekomma per observation, så kommer summan av förekomsterna vara större än antal observationer.



Figur 6: Förekomst av SNI-avdelningar hos köparen bland observationerna i datamaterialet.

Avdelningarna B, E, O, T och U förekommer inte alls. Detta betyder att de variabler som ger andelen av SNI-koder köparna har i dessa SNI-avdelningar kommer anta värdet 0 för alla observationer. Något man skulle kunna göra är att ta bort dessa variabler. Anledningen till att detta inte görs, är att modellen då inte kommer kunna appliceras på nya observationer där köparen faktiskt har en SNI-kod i någon av dessa avdelningar. Om

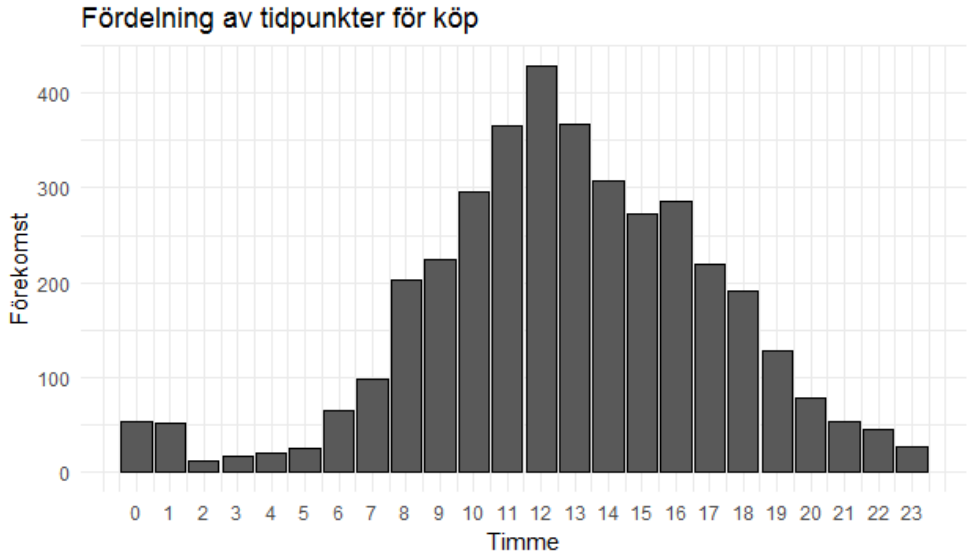
man instället låter dessa variabler vara kvar, kommer koefficienten för dessa variabler att bli 0 för alla delmodeller. För att se varför, låt oss titta på funktionen (9), som maximeras för att skatta parametrarna i modellen. Första delen utgörs av log-likelihood-funktionen. Eftersom variablerna för de fem icke-förekommande SNI-avdelningarna alltid är lika med 0, så kommer log-likelihood-funktionen vara helt oberoende av storleken hos koefficienterna för dessa variabler. Dock ser vi att regulariseringstermen blir som störst då alla dessa koefficienter är lika med 0. Slutsatsen blir då att (9) maximeras när koefficienterna för SNI-avdelningarna B, E, O, T och U är lika med 0 för alla delmodeller. I Figur 7 presenteras förekomsten av SNI-avdelningar hos säljare.



Figur 7: Förekomst av SNI-avdelningar hos säljarna bland observationerna i datamaterialet

3.2.2 Arbetstid och momskategori

Prediktorerna *arbetstid* och *momskategori* bygger på annan data som vi har tillgång till. Detta betyder att vi kan omdefiniera variablerna och se vilken av definitionerna som är bäst för vår modell. Den data som *arbetstid* bygger på är tidpunkten som köpet inträffade. Om det saknas tidpunkt för en observation, låter vi *arbetstid* anta värdet okänd/odefinierad. De resterande observationernas tidpunkter ges av Figur 8.



Figur 8: Fördelningen av vilken timme som köpet utfördes bland de observationer där information om tidpunkt fanns tillgänglig. Med timme x menas tidsintervallet från och med x.00 till och med x.59.

Vi kan nu definiera arbetsdag på olika sätt genom att variera tidpunkten när arbetsdag börjar och när en arbetsdag slutar. I modelleringen provas flera olika start- och sluttider.

Vi har en likande situation för moms. I Sverige har vi momssatserna 0, 6, 12 och 25%. I det datamaterial som används finns det endast information om den genomsnittliga momsen per händelse. Detta betyder att ifall den genomsnittliga momsen är låt oss säga 14%, så finns det många olika sätt som denna kan vara uppbyggd av momssatserna. Det sätt som prediktorn *momskategori* definieras är att om den genomsnittliga momsen är tillräckligt nära en av momssatserna, så sätts *momskategori* lika med denna momssats. Annars låter man prediktorn anta värdet okänd. Det man gör i modelleringen är att prova olika avstånd som bestämmer meningen i vad tillräckligt nära betyder i sammanhanget.

4 Statistisk modellering

4.1 Metod för statistisk modellering

Den modell vi ska anpassa är den multinomiala logistiska regressionsmodell (2) som beskrevs i teoriavsnittet. Applicerat på vårt klassificeringsproblem sätts $K = 19$, eftersom vi har 19 olika konton som responsvariabeln kan anta.

Det som görs i detta avsnitt är att bestämma vilka prediktorer som ska vara med, eventuella omdefinieringar av prediktorer samt skattningar av modellens parametrar med hjälp av regularisering. Den mängd data vi har delas upp i två delar så att 80% av datamängden är i träningsdelen och 20% av datamängden är i testdelen. Vid anpassning av en modell utförs korsvalidering. Anpassningen görs genom att maximera förlustfunktionen med en regulariseringsterm enligt funktion (9). Anpassat till vårt problem sätts $K = 19$ och $N = 6530$. Värdet på N sätts enligt hur många observationer som finns i träningsdata. Parametrarna anpassas genom att maximera funktionen för olika värden på regulariseringskonstanten λ. Innan modellen anpassas standardiseras prediktorerna, för att sedan transformeras tillbaka efter anpassning. Detta sker automatiskt i mjukvaran som används (Hastie and Qian, 2014). Dessutom får vi för varje λ som modellen anpassats på ett uppskattat värde på den multinomiala deviansen. Delar vi detta värde med 2 får vi värdet hos förlustfunktionen. Vi kommer då att välja den anpassning av modellen som ger lägst värde hos förlustfunktionen. Detta görs för varje modell som prövas. Jämförelser görs genom att jämföra förlustfunktionen som anpassningarna av modellerna gav upphov till där den modell med lägst värde på förlustfunktionen är att föredra.

4.2 Algoritm för modellselektion

Eftersom det finns oändligt många modeller vi skulle kunna anpassa, är det en bra idé att utgå från någon algoritm för modellselektion. Algoritmen som används är en blandning mellan forward selection och regularisering. Måttet som används för modellselektion är det genomsnittliga värdet på förlustfunktionen som fås genom korsvalidering. Algoritmen för modellselektion kan sammanfattas med hjälp av följande punkter.

När algoritmen har kört igenom huvudeffekterna så börjar en selektion av interaktionstermer. Denna selektion följer ungefär samma algoritm men har en del godtyckliga inslag.

1. Anpassa en modell som helt saknar prediktorer för olika värden på regulariseringskontanten. Välj den parameterskattning som ger lägst

genomsnittlig förlustfunktion.

2. Utifrån denna modell, anpassa alla modeller sådana att du endast lagt till en extra prediktor. Alla modeller anpassas med olika värden på regulariseringskonstanten och den parameterskattning som ger lägst genomsnittlig förlustfunktion väljs för varje enskild modell.

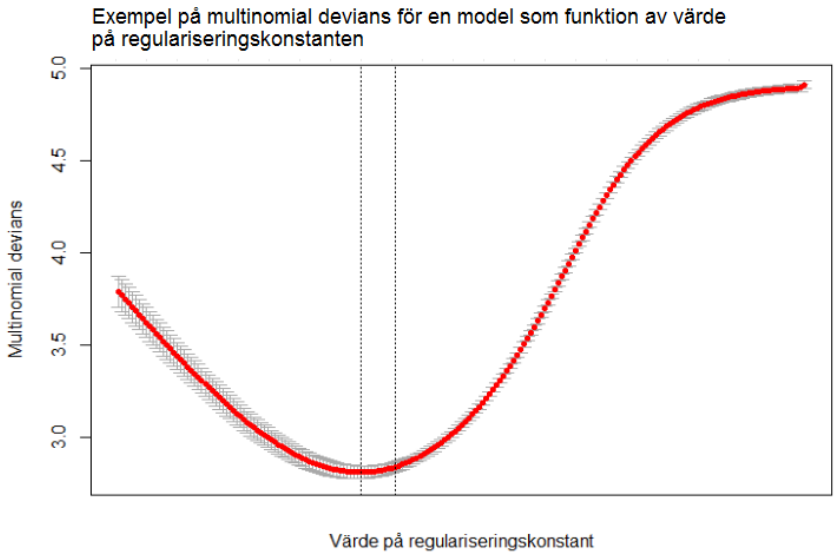
3. Jämför förlustfunktionerna hos den ursprungliga modellen och de nya modellerna. Den modell som har lägst förlustfunktion är den modell vi går vidare med. Om det visar sig vara den ursprungliga modellen så är detta vår slutgiltiga modell.

4. Om den modell vi valt i punkt (3) är en av de nya anpassade modellerna, så utgår vi från denna och går tillbaka till punkt (2). Ifall det inte finns fler resterande prediktorer så väljer vi den modell som innehåller alla prediktorer som slutmodell.

Vi gör ett ingrepp i algoritmen vid den tidpunkt den eventuellt ger konsekvensen att vi ska inkludera någon av variablerna *arbetstid* och *momskategori*. Ingreppet består av att vi lägger till variabeln i modellen och sedan provar olika versioner av hur variabeln är definierad. Anledningen till att vi gör detta är att dessa två variabler skapats från data vi har tillgång till. Detta gör det möjligt för oss att modifiera dessa. Dessutom finns det ett visst mått av godtycklighet i hur vi definierat dessa variabler. Variabeln *arbetstid* modifieras genom att först prova olika starttider för arbetstid, anpassa modellen för varje starttid, och sedan välja den starttid vars modell hade lägst förlustfunktion. När vi valt starttid gör vi samma sak för sluttiden för arbetstiden. Variabeln *momskategori* modifieras genom att vi ändrar avståndet mellan momssatsen och gränserna för vilka observationer som ska ingå i momssatsen. T.ex. om vi har satt avståndet till 2, så gäller det att alla observerade momssatser som befinner sig i intervallet $(23, 27)$ anges som momssats 0.25.

4.3 Mjukvara för modellselektion

Den mjukvara som används för modellselektionen är R-paketet *glmnet* (Friedman et al., 2010). Genom att använda funktionen *cv.glmnet()* anpassar vi en modell för en sekvens av olika värden på regulariseringskonstanten. Genom att plotta värdena på regulariseringskonstanten mot den multinomiala deviansen för de motsvarande anpassade modellerna kan vi se vilket värde på regulariseringskontanten som ger lägst värde hos förlustfunktionen. Ett exempel illustreras med Figur 9.



Figur 9: Ett exempel på värdet på multinomial devians för olika värden på regulariseringskonstanten. Den vänstra vertikala streckade linjen markerar det värde på regulariseringskonstanten som ger lägst multinomial devians. Den högra vertikala streckade linjen markerar det värde på regulariseringskonstanten vars skattning av multinomial devians har lägst standardavvikelse.

Paketet beräknar inte värdet hos förlustfunktionen, utan beräknar istället den multinomiala deviansen (MD). Såsom tidigare visats i teoriavsnittet finner vi värdet hos förlustfunktionen genom formeln

$$F(\boldsymbol{\alpha}, \boldsymbol{B}; \boldsymbol{X}, \boldsymbol{y}) = \frac{MD}{2}.$$

4.4 Anpassning av modellselektionen till mjukvara

På grund av begränsning i paketet *glmnet* behöver vi göra en mindre förändring i modellselektionsalgoritmen. Denna förändring innebär att vi inte anpassar modellen utan prediktorer eller modellerna som innehåller endast en av prediktorerna *pris, antal artiklar* och *arbetsdag*. Detta för att funktionen *cv.glmnet* kräver att modellen som anpassas ska innebära skattning av minst tre parametrar i varje delmodell som är inkluderad i den fullständiga modellen. Detta betyder att vi inte kan använda funktionen för att anpassa dessa modeller. Detta ses inte som ett stort problem eftersom ingen av dessa modeller förväntas ha den bästa prediktionsförmågan.

4.5 Modellselektion

I Tabell 3 sammanfattas första delen av modellselektionen där endast huvudeffekter ingår. Vi ser att vi först lägger till variabeln *säljare_sni*, sedan

	Ingående variabler	Värde på förlustfunktion
Modell 1	säljare_sni	1.7858
Modell 2	säljare_sni, köpare_sni	1.6618
Modell 3	säljare_sni, köpare_sni, momskategori	1.5765
Modell 4	säljare_sni, köpare_sni, momskategori	1.5765
Modell 5	säljare_sni, köpare_sni, momskategori, arbetstid	1.5373
Modell 6	säljare_sni, köpare_sni, momskategori, arbetstid_korr	1.5417
Modell 7	säljare_sni, köpare_sni, momskategori, arbetstid_korr	1.5417
Modell 8	säljare_sni, köpare_sni, momskategori, arbetstid_korr, pris	1.5094
Modell 9	säljare_sni, köpare_sni, momskategori, arbetstid_korr, pris, antal_artiklar	1.5001
Modell 10	säljare_sni, köpare_sni, momskategori, arbetstid, pris, antal_artiklar, betalningssätt	1.4919

Tabell 3: En tabell över den inledande modellselektionen med enbart huvudeffekter.

köpare_sni och därefter *momskategori*. Därefter modifieras modell 3 genom att prova olika versioner av variabeln *momskategori*. Vi kommer fram till att hur momskategori redan var definierad, dvs. att vi sätter avståndet lika med 1, är det som ger lägst värde på förlustfunktionen. På grund av detta är modell 3 och 4 densamma. Vi återupptar selektionsalgoritmen och resultatet blir att lägga till variabeln *arbetstid*. Modell 6 är resultatet av att vi provat olika starttider för variabeln *arbetstid* i modell 5. Vi kommer fram till att vi får lägre värde på förlustfunktionen om vi ändrar starttiden från 08.00 till 09.00 och därav har vi ersatt *arbetstid* med *arbetstid_korr* i modell 6. I modell 7 provar vi olika sluttider för arbetstidsvariabeln. Vi kommer fram till den urspungliga sluttiden, 17.00. Detta betyder att modell 6 och 7 är samma modeller. Något att notera är att värdet på förlustfunktionen i modell 6 och 7 är större än i modell 5. Detta har att göra med att funktionen

cv.glmnet() använder sig av slump då den delar upp träningsdelen i fem olika delar. Eftersom delarna då inte blir identiska kan resultaten skilja sig vid olika tillfällen när man anropar funktionen på samma dataset. Modell 8 till 10 är resultat av att vi fortsatt med selektionsalgoritmen. Vi finner att alla variabler har tagits med förutom variabeln *arbetsdag*.

Modelleringen fortskrider med att inkludera interaktionstermer. Tabell 4 visar de modeller som var kandidater till att bli modell 1. Vi ser att va-

Ingående variabler	Värde på förlustfunktion	95% konfidensintervall för förlustfunktion
betalningssätt	2.4178	(2.4060, 2.4296)
köpare_sni	2.2589	(2.2504, 2.2673)
säljare_sni	1.7858	(1.7708, 1.8009)
momskategori	2.2230	(2.2081, 2.2380)
arbetstid	2.3107	(2.3060, 2.3154)

Tabell 4: Tabell över förlustfunktionsvärdena hos de modeller som var kandidater till att bli modell 1. 95% konfidensintervall för förlustfunktionsvärdena har även inkluderats.

riabeln *sni_säljare* var den prediktor som fick lägst förlustfunktionsvärde. Utifrån konfidensintervallen kan vi dra slutsatsen att skillnaden även är statistiskt signifikant. På grund av detta ansågs att det vara intressant att utföra samma algoritm som på huvudeffekterna, men för de interaktionstermer där *sni_säljare* ingår. Eftersom huvudeffekten hos *arbetstid* inte kom med provar vi inte ens interaktioner som innehåller denna variabel. Tabell 5 sammanfattar denna del av modellselektionen. Bortsett från interaktionstermerna utgörs modellerna även av alla de huvudeffekter som ingår i modell 10. I modell 15 ingår alla interaktionstermer med *säljare_sni* förutom med variablerna *antal_artiklar* och *arbetstid*.

	Ingående interaktionstermer med säljare_sni	Värde på förlustfunktion
Modell 11	köpare_sni	1.4805
Modell 12	köpare_sni, momskategori	1.4583
Modell 13	köpare_sni, momskategori, arbetstid	1.4417
Modell 14	köpare_sni, momskategori, arbetstid, pris	1.4354
Modell 15	köpare_sni, momskategori, arbetstid, pris, betalningssätt	1.4335

Tabell 5: En tabell över modellselektionen för interaktionstermer med prediktorn *säljare_sni*.

	Ingående interaktionstermer med köpare_sni	Värde på förlustfunktion
Modell 16	säljare_sni, momskategori	1.4205
Modell 17	säljare_sni, momskategori, arbetstid	1.4102
Modell 18	säljare_sni, momskategori, arbetstid, pris	1.4051

Tabell 6: En tabell över modellselektionen för interaktionstermer med prediktorn *köpare_sni*.

Modellselektionen fortsätter sedan genom att titta på interaktioner där variabeln *köpare_sni* ingår. Resultatet sammanfattas av Tabell 6. Notera att interaktionen mellan *köpare_sni* och *säljare_sni* ingår sedan tidigare. Alla andra prediktorer som återfinns i modell 15 finns med i modell 16-19.

Nu har vi med alla interaktionstermer där *säljare_sni* ingår förutom interaktionerna med *antal_artiklar*, *betalningssätt* och *arbetsdag*. Vinsterna i att lägga till fler interaktionstermer är nu så små och därför prövas en sista modell där interaktionstermen mellan *arbetstid* och *momskategori* har lagts till. Det visar sig bli en liten förbättring och därför låter vi denna modell bli vår slutgiltiga modell. Denna modell (modell 19) sammanfattas av Tabell 7.

4.6 Slutgiltig modell

Här presenteras endast formen på den slutgiltiga modellen och inte de exakta koefficienterna.

	Ingående prediktorer	Värde på förlustfunktion
Modell 19	Alla prediktorer i modell 18, arbetstid:momskategori	1.4044

Tabell 7: Sammanfattning av den slutgiltiga modellen.

$$\begin{aligned}\log \frac{p_k(\boldsymbol{x})}{p_{19}(\boldsymbol{x})} =& \alpha_k + \beta_{k1} x_{pris} + \beta_{k2} x_{antal\_artiklar} + \beta_{k,betalningsatt} + \beta_{k,kopare\_sni} \\ &+ \beta_{k,saljare\_sni} + \beta_{k,momskategori} + \beta_{k,arbetstid} + \gamma_{k,kopare\_sni} x_{pris} \\ &+ \gamma_{k,kopare\_sni:saljare\_sni} + \gamma_{k,kopare\_sni:momskategori} \\ &+ \gamma_{k,kopare\_sni:arbetstid} + \gamma_{k,saljare\_sni} x_{pris} + \gamma_{k,saljare\_sni:betalningssatt} \\ &+ \gamma_{k,saljare\_sni:momskategori} + \gamma_{k,saljare\_sni:arbetstid} \\ &+ \gamma_{k,momskategori:arbetstid}, \\ & k = 1, 2, \ldots 18\end{aligned}$$

Indexeringen k framhåller att dessa termer är specifika för den k:te modellen. Den första termen som noteras med α_k är interceptet. De termer som noteras med β är effekter från endast en prediktor. Utav dessa termer, är de som inte innehåller någon faktor benämnd x skrivna i faktorformat. T.ex. antar $\beta_{k,momskategori}$ olika värden beroende på vilken nivå denna faktor antog. För att modellens parametrar ska vara entydigt bestämda finns det en nivå i varje faktorvariabel vars motsvarande term i modellen alltid antar värdet 0. De termer som noteras med γ är interaktionseffekter. För entydighet hos modellens parametrar krävs det att om någon av de ingående faktorerna i en interaktionsterm antar basfallet, så kommer interaktionstermen att vara lika med 0.

5 Modellutvärdering

5.1 Exakthet

Den kanske mest naturliga frågan som man vill ha reda på är hur ofta modellen får rätt. Genom att låta modellen prediktera på testdata, vilket är oberoende av den data som användes för att anpassa modellen, får vi en exakthet på 57.02%. För att jämföra, så får man en exakthet på 5.26% om

man predikterar genom att likformigt slumpa ett konto och en exakthet på 18.41% om man gissar på det mest förekommande kontot.

Exakthet som mått beror på vilken data som råkar hamna i testdelen. Det mått vi egentligen är intresserad av är förväntad exakthet. Vi kan med hjälp av exakthet beräkna ett 95% konfidensintervall för förväntad exakthet. Om vi multiplicerar Acc med antalet observationer i testdata, N, får vi antalet observationer som predikterades rätt. Vi utgår från antagandet om att vi inte kan förespå vilka värden som prediktorerna antar för varje observation. Från detta utgångsläge är sannolikheten för att den i:te observationen predikteras rätt densamma som för den j:te observationen för alla $i, j = 0, \ldots, N$. Detta påstående är rimligt eftersom vi inte vet om den i:te eller j:te observationer kommer ha värden som ger säkrare prediktioner eller inte. Vi utgår även från antagandet om att observationerna i testdata är oberoende. Antalet observationer som predikteras rätt ges därmed av en summa av likafördelade och oberoende Bernoulli-slumpvariabler, vilket är detsamma som den binomialfördelade slumpvariabeln

$$p(x) = \binom{N}{x} \theta^x (1-\theta)^{N-x} \quad x = 0, 1, \ldots, N.$$

där θ representerar $E[Acc]$. Det konfidensintervall som vi hittar för θ blir därmed detsamma som konfidensintervallet för $E[Acc]$. Om vi utgår från (12) får vi Wald-konfidensintervallet

$$CI_{WALD} = (0.5461, 0.5943).$$

5.2 Modellen som stöd för bokföring

Modellen kan användas som ett beslutstöd i bokföring. Då vill man att modellen endast ger förslag då den är relativt säker i sin prediktion. Om vi låter modellen endast prediktera när den skattade sannolikheten för det predikterade kontot är minst 0.9 så predikterar modellen 11.27% av gångerna. Detta skulle betyda att förutsatt att sannolikheten 0.9 är tillfredställande, kan modellen ge beslutsstöd för uppskattningsvis runt 11% av bokföringshändelserna. Notera att modellen även känner av vilka bokföringshändelser som den ger rekommendationer för.

När vi har tillgång till testdata, bör vi inte ta för givet att modellen verkligen predikterar med 90% säkerhet för dessa observationer, utan vi bör även testa detta. Vi kan nu beräkna exakthet endast för de observationer som enligt modellen kan predikteras med minst 90% säkerhet. Exakthet beräknas till 95.08% vilket tyder på att modellen verkligen predikterar med denna säker-

het. Analogt med föregående avsnitt kan vi konstruera ett konfidensintervall för den förväntade exaktheten. Denna ges av

$$CI_{WALD} = (0.9195, 0.9822)$$

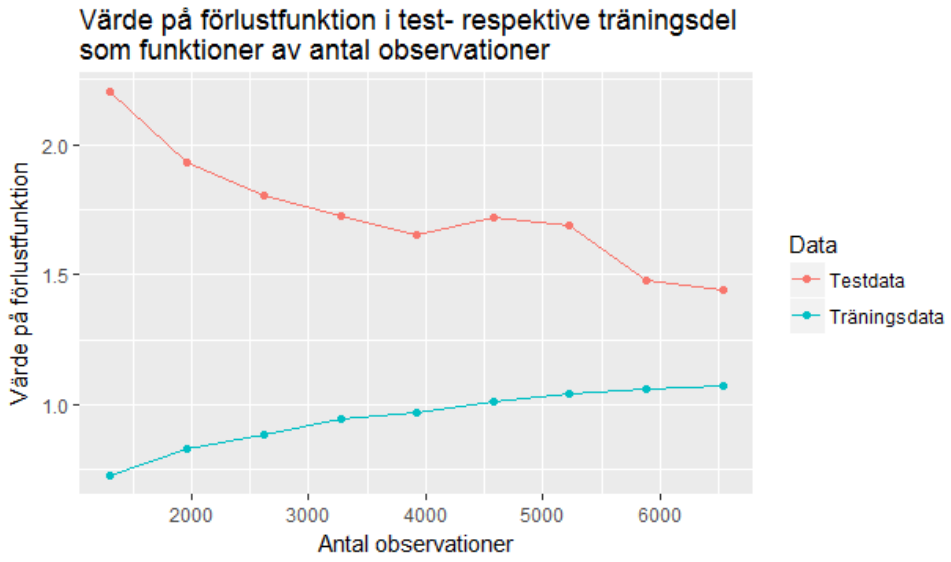
5.3 Jämförelse av förlustfunktioner

När man provar olika modeller i träningsdelen, finns det risk att man väljer en modell som av slump ger en bra anpassning för alla datapunkter i träningsdata. Om detta händer, upptäcker vi inte detta genom korsvalidering. Ett sätt att upptäcka detta är att beräkna förlustfunktionen i testdelen. Med helt ny data är sannolikheten att även denna data av slump skulle passa väl till modellen liten. Förlustfunktionen i testdelen beräknas till 1.4426 vilket är något större än 1.4051 som förlustfunktionen blev då vi använde korsvalidering i träningsdata. Vi bör också ta hänsyn till att när vi utförde korsvalidering använde vi endast $\frac{4}{5}$ av träningsdelen i anpassningarna som användes för att beräkna förlustfunktionen. När vi beräknar förlustfunktionen i testdelen använder vi hela träningsdelen. Detta talar för att förlustfunktionen borde vara lite mindre för testdelen, givet att vi inte har den slumpeffekt som beskrevs tidigare. Trots detta anses 1.4426 vara tillräckligt nära för att vi ska anse att korsvalideringen valt ut en väl presterande modell i jämförelse med de andra modellerna vi hade att välja mellan.

5.4 Analys med inlärningskurvor

Träningsdata delades upp i tio delar. Därefter anpassades modellen på de två första delarna. Regulariseringskonstanten justeras så att den är lika med konstanden i den slutgiltiga modellen multiplicerad med hur stor andel observationer som den data modellen anpassas på utgör av hela träningsdata. Detta gör så att styrkan i regulariseringen är densamma vid alla anpassningar. Därefter utökades data med nästa del och modellen anpassades återigen. Detta upprepades tills data hade utökats till att omfatta hela träningsdata. Modellen har då anpassats 9 gånger på data av olika storlekar. För alla dessa anpassningar beräknades förlustfunktionen på testdelen samt förlustfunktionen då den beräknas på den data som modellen anpassats till. Detta leder till Figur 10.

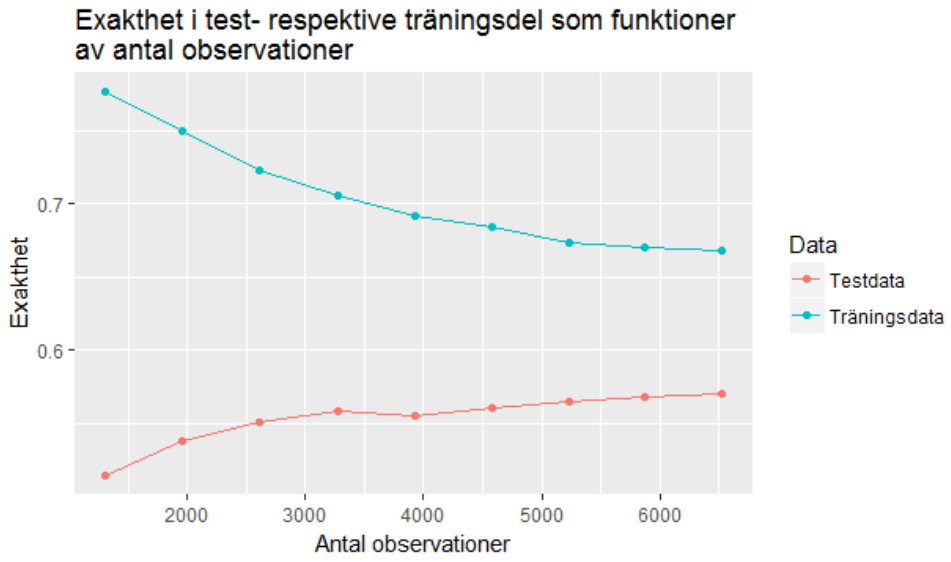
Kurvorna i Figur 10 indikerar att mer data säkerligen skulle leda till en bättre modell. Dels har vi ett stort gap mellan värdet hos förlustfunktionen för test respektive träningsdata i den sista punkten, dels ser det ut som att förlustfunktionsvärdet för testdata minskar snabbare än det ökar för träningsdata. Uppskattningsvis är det möjligt att komma ner till ett värde på 1.25.



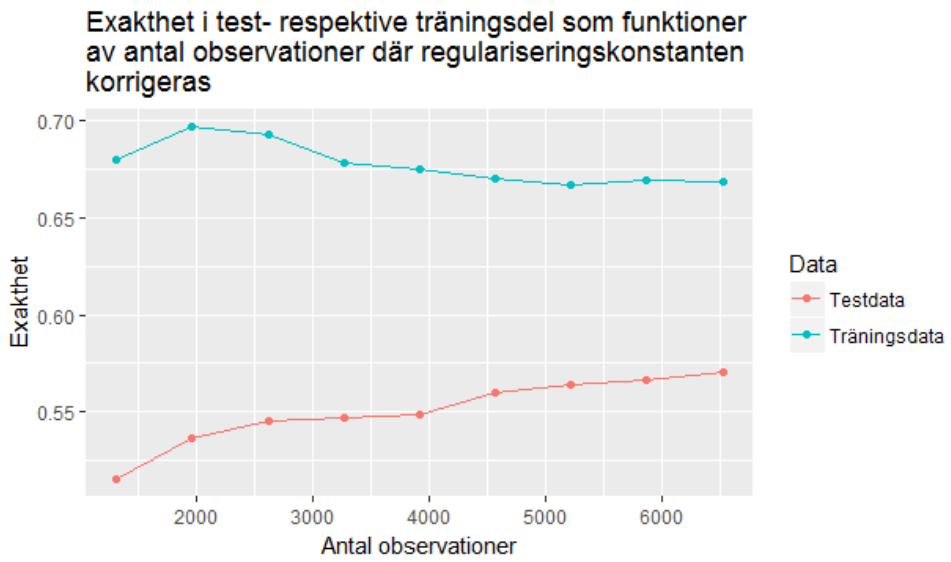
Figur 10: Förlustfunktionsvärdet för träningsdata respektive testdata som funktioner av hur många observationer som finns i träningsdata.

Värdet hos förlustfunktionen applicerad på hela träningsdata är enligt figuren 1.07. Det bör vara omöjligt att hamna under detta utan att lägga till eller byta ut prediktorer, alternativt ändra värdet på regulariseringskonstanten. För att få en känsla av vad detta betyder, plottas i Figur 11 inlärningskurvor där vi istället för förlustfunktionsvärdet använder oss av exakthet.

Förlustfunktionsvärdet 1.07 motsvaras i detta fall enligt Figur 11 av en modell som har 66,8% i exakthet. Om vi anser att vi inte är nöjda med ett sådant värde på exakthet bör vi komplettera åtgärden att skaffa mer data med andra sätt att förbättra modellen. Ett alternativ är att ändra regulariseringskonstanten. I Figur 12 plottas en inlärningskurva med exakthet som mått, men där regulariseringskonstanten har korrigerats i varje anpassning med hjälp av korsvalidering.



Figur 11: Exakthet för träningsdata respektive testdata som funktioner av hur många observationer som finns i träningsdata.



Figur 12: Exakthet för träningsdata respektive testdata som funktioner av hur många observationer som finns i träningsdata, när regulariseringskonstanten korrigerats vid varje anpassning för att minimera förlustfunktionen i korsvalidering.

Tanken med Figur 12 att detta ska ge oss en uppfattning om hur ändring av regulariseringskonstanten i samband med ökat datamaterial kan förbättra modellen.

Vi ser i Figur 12 att exakthet beräknat på testdata börjar med att öka något för att minska långsamt med antalet observationer. Om vi antar att den minskande trenden kommer att fortsätta när vi ökar antalet observationer, betyder detta att vi inte kan räkna med en modell som är bättre än drygt 65% exakthet även om vi ökar mängden data och samtidigt korrigerar regulariseringskonstanten.

Enligt den analys som gjorts med inlärningskurvor skulle modellen bli bättre med en ökad mängd data. Utöver detta krävs det nya prediktorer som kan särskilja klasserna i responsvariabeln på ett bättre sätt än de nuvarande prediktorerna gör.

5.5 Felmatris

I Figur 13 presenteras felmatrisen för hur modellen har presterat på testdata. Elementet på rad i och kolumn j säger oss hur många observationer som bokfördes på konto i men predikterades enligt modellen som konto j. Diagonalen är grönmarkerad och anger hur många händelser som predikterats rätt per konto.

5.6 Modellens prestation för de enskilda kontona

För att mäta hur väl modellen presterar på de enskilda kontona använder vi måttet känslighet. Känsligheten för varje enskilt konto presenteras i Figur 14.

Vi ser i Figur 14 att inga observationer som tillhör konto 12, 14 och 19 predikteras rätt. Om vi vill att modellen ska prestera lika bra oberoende av vilket konto som observationen till slut hamnar, kan det vara en god idé att hitta prediktorer som kan särskilja konton förknippade med lågt känslighetsmått gentemot de andra kontona. Som ett komplement till exakthet kan vi använda oss av felmatrisen som presenterades i föregående avsnitt. Därifrån kan vi utläsa vilka konton som predikterades istället för att prediktera det rätta kontot. Vi ser t.ex. att för de observationer där kontot som bokfördes var konto 12, predikterade modellen två gånger konto 1, två gånger konto 6 och en gång konto 10. För att få modellen att prediktera konto 12, skulle introducerandet av en prediktor som kunde skilja konto 12 från dessa konton vara idealt.

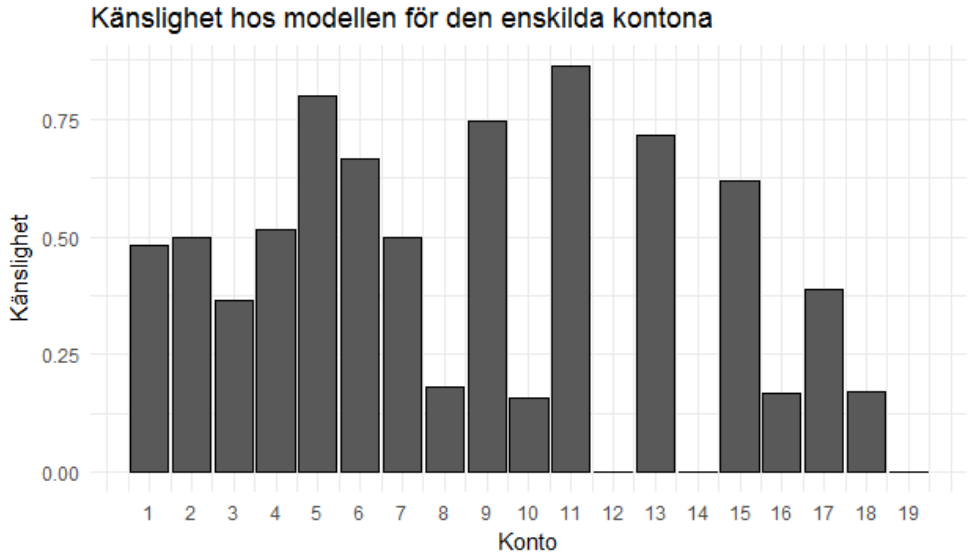
Ifall vi är främst intresserade av att hitta sätt att öka exakthet, är det bättre att hitta prediktorer som särskiljer konton som i absoluta tal har flest

$$\begin{bmatrix}
76 & 0 & 0 & 0 & 0 & 61 & 9 & 0 & 1 & 0 & 4 & 0 & 0 & 0 & 4 & 1 & 0 & 1 & 1 \\
1 & 3 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 4 & 1 & 0 & 2 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \\
1 & 0 & 0 & 33 & 0 & 7 & 1 & 0 & 1 & 0 & 0 & 0 & 7 & 0 & 9 & 3 & 2 & 0 & 0 \\
0 & 0 & 0 & 0 & 4 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
36 & 0 & 0 & 3 & 0 & 199 & 11 & 0 & 9 & 2 & 6 & 0 & 12 & 0 & 8 & 12 & 0 & 1 & 0 \\
9 & 0 & 1 & 2 & 0 & 30 & 79 & 0 & 19 & 0 & 1 & 0 & 6 & 0 & 2 & 6 & 3 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 2 & 4 & 0 & 0 & 0 & 4 & 0 & 0 & 0 & 0 & 0 & 0 \\
2 & 0 & 0 & 1 & 0 & 12 & 9 & 0 & 159 & 1 & 18 & 0 & 2 & 0 & 0 & 9 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 6 & 0 & 0 & 0 & 7 & 5 & 0 & 20 & 0 & 4 & 0 & 0 & 1 & 0 \\
2 & 0 & 0 & 1 & 0 & 5 & 3 & 0 & 3 & 0 & 133 & 0 & 0 & 0 & 2 & 3 & 0 & 2 & 0 \\
2 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 1 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 20 & 3 & 0 & 5 & 1 & 1 & 0 & 126 & 0 & 17 & 2 & 0 & 0 & 0 \\
1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 4 & 1 & 0 & 0 \\
2 & 0 & 0 & 2 & 0 & 4 & 3 & 0 & 1 & 0 & 2 & 0 & 16 & 0 & 65 & 9 & 1 & 0 & 0 \\
4 & 0 & 0 & 3 & 0 & 27 & 13 & 0 & 10 & 1 & 11 & 0 & 19 & 2 & 6 & 21 & 1 & 8 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 7 & 7 & 0 & 0 \\
3 & 0 & 0 & 0 & 0 & 6 & 1 & 0 & 6 & 0 & 11 & 0 & 2 & 1 & 3 & 6 & 0 & 8 & 0 \\
0 & 0 & 0 & 1 & 0 & 2 & 0 & 0 & 1 & 0 & 1 & 0 & 3 & 0 & 1 & 3 & 1 & 0 & 0
\end{bmatrix}$$

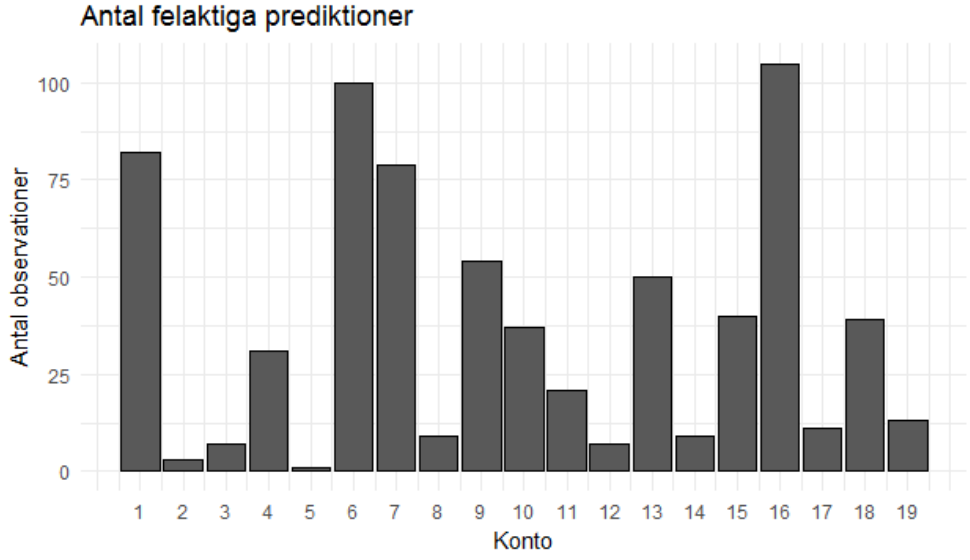
Figur 13: Felmatris för den slutgiltiga modellen applicerad på testdata. Elementet på rad i och kolumn j anger antalet observationer som bokfördes på konto i och predikterades på konto j.

observationer som blir felaktigt predikterade. Då är det bättre att utgå från Figur 15.

Om vi nu tittar på konto 12, 14 och 19 i Figur 15 så är de bland kontona som är av lägst intresse från ett perspektiv där vi vill öka exaktheten med så mycket som möjligt. Detta har att göra med att dessa konton förekommer sällan. Istället är det kontona 1, 6, 7 och 16 som vi vill öka prediktionsförmågan för.



Figur 14: Känsligheten hos de enskilda kontona när modellen appliceras på testdata.



Figur 15: Antal felaktiga prediktioner per konto då modellen appliceras på testdata.

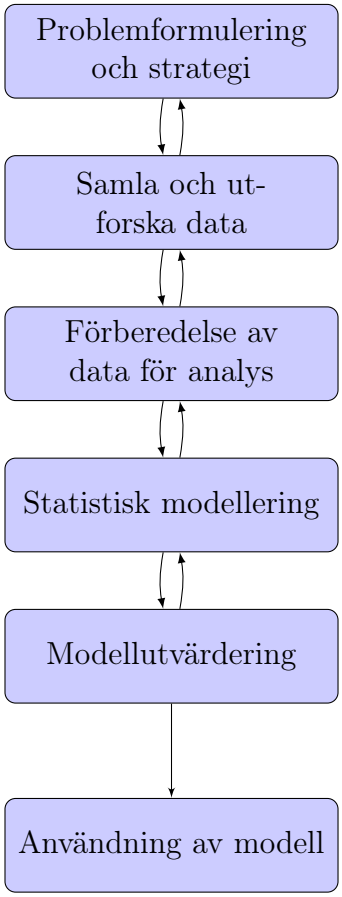
6 Resultat och diskussion

För att sammanfatta resultaten har vi nu en modell med uppskattad exakthet på 57.02%. Med hjälp av måttet känslighet har de kontona som modellen presterar speciellt dåligt på identifierats. Ett bra sätt att förbättra modellen vore att hitta prediktorer som på ett bra sätt kan särskilja dessa konton från andra konton. Analysen med inlärningskurvor leder oss till slutsatsen att om vi ska få en modell med hög exakthet bör vi öka både mängden data och introducera nya prediktorer.

Ett annat alternativ är att byta modelltyp. Ett problem med att använda en generalisering av logistisk regression är att om man vill fånga effekten av interaktioner på responsvariabeln eller kvadratiska relationer mellan prediktorer och responsvariabeln, blir det snabbt väldigt många parametrar att skatta i delmodellerna. Detta gäller speciellt om man vill introducera en stor mängd av nya prediktorer. För att fånga icke-lineära samband är det bättre att använda icke-lineära metoder. Två vanliga icke-linjära klassificeringsmetoder är *support vector machine* och *artificiella neurala nätverk*. Gemensamt för båda metoderna är de resulterande modellerna först transformerar rummet där vektorerna som ger värdena för prediktorerna är definierade. Det gäller alltså att $\mathbf{x} = (x_1, \ldots, x_p)$ transformeras till $\mathbf{f} = (f_1(\mathbf{x}), \ldots, f_k(\mathbf{x}))$. Därefter kan metoderna ses som linjära metoder på dessa nya prediktorer som ges av $\mathbf{f}$ (Hastie et al. 2016 s.392, 417).

Eftersom detta är ett kandidatarbete i matematisk statistik, var arbetet främst inriktat på att de statistiska aspekterna i att bygga en modell. Man skulle säkerligen ha fått bättre resultat om man hade använt ett bredare angreppssätt. Ett mycket vanligt angreppssätt inom industrin är att använda sig av *CRISP-DM*, som står för *The cross-industry standard process for data mining* (Larose and Larose, 2015, s.6). I Figur 16 presenteras denna som ett flödeschema. Vad *Problemformulering och strategi* är säger sig självt, men här ingår även att värdera vilket ekonomiskt värde som kan tänkas komma ut från projektet. När det kommer till att samla och utforska data, ingår det att avgöra kvalitén hos data. Detta gjordes inte i denna uppsats, och är mycket viktigt för att få rättvisande resultat. Mer tid för att hitta fler passande prediktorer hade även det troligtvis förbättrat resultaten. Huvudfokus i uppsatsen låg på *Statistisk modellering* och *Modellutvärdering*. Något att notera i Figur 16 är att flödet även går bakåt i alla noder förutom i den sista. Om man t.ex. upptäcker något i modellutvärderingen som man tror kan åtgärdas kan man gå tillbaka till modelleringen. Ett sådant arbetssätt hade kunnat användas mer i uppsatsen. Speciellt hade måtten för känslighet för varje konto kunnat användas som komplement till förlustfunktioner i selektionen av

modeller. Då hade känsligheten beräknats genom korsvalidering.



Figur 16: Ett flödesschema enligt CRISP-DM.

Trots att modellen är långt ifrån att kunna fungera som underlag för automatiserad bokföring, bör man kunna känna sig hoppfull över att det faktiskt är möjligt att skapa en sådan modell. Som påpekats finns det flera aspekter för förbättring. Det som ändå talar främst för att det är möjligt med automatiserad bokföring är följande resonemang. När en person som arbetar med bokföring ska bokföra en kostnad använder denne en viss mängd information, t.ex. det som finns på kvittot eller vad bokföraren vet om företagen som är inblandade i händelsen. Vi kan se detta som att bokföraren utgår från en internaliserad modell för vilket konto som ska bokföras, och att denna internaliserade modell är så gott som helt felfri. Låt oss kontrastera detta mot att handla med aktier eller prediktera resultatet i fotbollsmatcher. Även de som är mycket duktiga på att prediktera dessa fenomen har ofta fel.

En aktiehandlare som presterade på samma nivå som bokföraren gör när det kommer till bokföring, skulle antagligen anses vara synsk.

Eftersom bokförare använder relativt begränsad mängd information och kan säga med så stor sannolikhet vilket konto som en kostnad bör bokföras på, kan man anta att det finns prediktorer som, givet att den antar vissa värden, nästan helt flyttar sannolikhetsmassan till en del av utfallsrummet. Vad detta betyder är att i vissa fall fungerar prediktorerna nästan som logiska satser sådana att om de antar ett visst värde, så vet vi nästan helt säkert att kontot kommer tillhöra en viss äkta delmängd av mängden av alla konton. Ett exempel skulle kunna vara ifall vi hade en prediktor som säger ifall ordet *bensin* förekommer på bokföringsunderlaget eller inte. För de observationer där ordet bensin förekommer, kan vi nog vara relativt säkra på att det är en resekostnad vi ska bokföra. Att hitta dessa prediktorer, vilka säkert är väldigt många, kommer troligtvis vara nyckeln till en väl presterande modell för prediktion av kostnadskonto.

Bilagor

A Härledning av förlustfunktion

Vi antar att observationerna vi har är oberoende. Detta betyder att likelihood-funktionen ges av produkten av likelihood-funktionerna för alla enskilda observationer. Låt oss nu titta på slumpvariabeln för den i:te observationen som vi benämner Y_i. Vi låter värdena som prediktorerna antar för denna observation ges av vektorn $\boldsymbol{x}_i$. Enligt modellen ges fördelningen för Y_i av den multinomiala fördelningen.

$$\begin{aligned} p(n_1, n_2, \dots n_K; \boldsymbol{x}_i) &= \frac{1!}{n_1! n_2! \cdots n_{K-1}! n_K!} p_1(\boldsymbol{x}_i)^{n_1} p_2(\boldsymbol{x}_i)^{n_2} \cdots p_K(\boldsymbol{x}_i)^{n_K} \\ &= p_1(\boldsymbol{x}_i)^{n_1} p_2(\boldsymbol{x}_i)^{n_2} \cdots p_K(\boldsymbol{x}_i)^{n_K} \end{aligned}$$

där n_k är lika med 1 om Y_i antar kategori k, och 0 annars. Vi skulle kunna skriva fördelningen i termer av $K-1$ parametrar men detta sätt att representera blir tydligare. Låt oss säga att Y_i antog värdet k, dvs. $y_i = k$. Då ges likelihood-funktionen av

$$\begin{aligned} \mathcal{L}(\boldsymbol{\alpha}, \boldsymbol{B}; \boldsymbol{x}_i, y_i) &= p_1(\boldsymbol{x}_i)^0 \cdots p_k(\boldsymbol{x}_i)^1 \cdots p_K(\boldsymbol{x}_i)^0 \\ &= p_k(\boldsymbol{x}_i) \end{aligned}$$

Likelihood-funktionen för alla observationer ges då av

$$\mathcal{L}(\boldsymbol{\alpha}, \boldsymbol{B}; \boldsymbol{X}, \boldsymbol{y}) = \prod_{i=1}^{N} \sum_{k=1}^{K} y_{ik} p_k(\boldsymbol{x}_i),$$

där y_{ik} är lika med 1 om den i:te observationen antar värdet k, och annars 0. Vektorn $\boldsymbol{y}$ är de observerade utfallen hos responsvariabeln. Log-likelihood-funktionen fås genom att ta logaritmen av likelihood-funktionen. Log-likelihood-funktionen ges av

$$
\begin{aligned}
\ell(\boldsymbol{\alpha}, \boldsymbol{B}; \boldsymbol{X}, \boldsymbol{y}) &= \log\left(\prod_{i=1}^{N}\sum_{k=1}^{K} y_{ik} p_k(\boldsymbol{x}_i)\right) \\
&= \sum_{i=1}^{N} \log\left(\sum_{k=1}^{K} y_{ik} p_k(\boldsymbol{x}_i)\right) \\
&= \sum_{i=1}^{N}\sum_{k=1}^{K} \log\left(y_{ik} p_k(\boldsymbol{x}_i)\right) \\
&= \sum_{i=1}^{N}\sum_{k=1}^{K} y_{ik} \log p_k(\boldsymbol{x}_i)
\end{aligned}
$$

Viktigt att notera är att då vi för in logaritmeringen innanför summeringen över alla k, så kan vi göra detta eftersom det endast finns en nollskild term i denna summering. Något annat som är viktigt att hålla reda på är att termerna $y_{ik} \log p_k(\boldsymbol{x}_i)$ där den i:te observationen inte antar värdet k ska tolkas som 0. Vi kan nu använda uttrycket för sannolikheterna $p_k(\boldsymbol{x}_i)$ som ges av (5).

$$
\begin{aligned}
\ell(\boldsymbol{\alpha}, \boldsymbol{B}; \boldsymbol{X}, \boldsymbol{y}) &= \sum_{i=1}^{N}\sum_{k=1}^{K} y_{ik} \log\left(\frac{\exp\left(\alpha_k + \boldsymbol{\beta}_k^T \boldsymbol{x}_i\right)}{\sum_{i=1}^{K} \exp\left(\alpha_k + \boldsymbol{\beta}_k^T \boldsymbol{x}_i\right)}\right) \\
&= \sum_{i=1}^{N}\sum_{k=1}^{K} y_{ik} \left(\log\left(\exp\left(\alpha_k + \boldsymbol{\beta}_k^T \boldsymbol{x}_i\right)\right) - \log\left(\sum_{i=1}^{K} \exp\left(\alpha_k + \boldsymbol{\beta}_k^T \boldsymbol{x}_i\right)\right)\right) \\
&= \sum_{i=1}^{N} \left(\sum_{k=1}^{K} y_{ik} \log\left(\exp\left(\alpha_k + \boldsymbol{\beta}_k^T \boldsymbol{x}_i\right)\right) - \log\left(\sum_{i=1}^{K} \exp\left(\alpha_k + \boldsymbol{\beta}_k^T \boldsymbol{x}_i\right)\right)\right)
\end{aligned}
$$

Vi har här utnyttjat den kända regeln $\log \frac{a}{b} = \log a - \log b$ och att oberoende av vilket värde som responsvariabeln antar vid den i:te observationen, är täljaren i kvoten innanför logaritmfunktionen lika med $\sum_{i=k}^{K} \exp\left(\alpha_k + \boldsymbol{\beta}_k^T \boldsymbol{x}_i\right)$. Delar vi uttrycket för log-likelihood-funktionen med $-\frac{1}{N}$ får vi förlustfunktionen vi uttryckt i (7).

B Studie av partiell derivata av förlustfunktionens i ett specialfall

I avsnittet om regularisering togs det upp ett fall där de observationer där prediktorn x antog värdet 1 endast bestod av sådana som antog en klass c.

Prediktorn antog annars värdet 0. Vi ska nu visa att den delsumma av log-likelihood-funktionen som prestenterades är strikt växande med avseende på parametern β_x som är parametern som bestämmer vilken påverkan som x har på kvoten mellan sannolikheten för klass c och sannolikheten för basfallet. Summanden i (10) kan delas upp i en yttre och två inre funktioner.

$$\begin{aligned} f_1(x) &= \log x \\ f_2(x) &= \frac{x}{\sum_{i=1}^{K} I(k \neq c) \exp\left(\alpha_k + \boldsymbol{\beta}_k^T \boldsymbol{x}_i\right) + x} \\ f_3(x) &= \exp\left((\alpha_c + \bar{\boldsymbol{\beta}}_c^T \bar{\boldsymbol{x}}_i + x)\right) \end{aligned}$$

Anledningen till att det nu finns ett streck ovanför $\boldsymbol{\beta}$ och $\boldsymbol{x}$ är att vi tagit bort β_x och värdet på prediktorn x från vektorerna. Låter vi sammansättningen vara

$$f_1 \circ f_2 \circ f_3$$

och sätter $x = \beta_x$ i f_3 så kommer vi få summanden i (10). Med hjälp av den sammansatta funktionen och kedjeregeln kan vi beräkna den partiella derivatan med avseende på β_x. Vi låter $g = f_1 \circ f_2 \circ f_3$.

$$\begin{aligned} \frac{\partial g}{\partial \beta_x} =& \frac{\partial f_1}{\partial f_2} \cdot \frac{\partial f_2}{\partial f_3} \cdot \frac{\partial f_3}{\partial \beta_x} \\ =& \frac{\sum_{i=1}^{K} \exp\left(\alpha_k + \boldsymbol{\beta}_k^T \boldsymbol{x}_i\right)}{\exp\left(\alpha_c + \bar{\boldsymbol{\beta}}_c^T \bar{\boldsymbol{x}}_i + \beta_x\right)} \cdot \frac{\sum_{i=1}^{K} I(k \neq c) \exp\left(\alpha_k + \boldsymbol{\beta}_k^T \boldsymbol{x}_i\right)}{\left(\sum_{i=1}^{K} \exp\left(\alpha_k + \boldsymbol{\beta}_k^T \boldsymbol{x}_i\right)\right)^2} \cdot \\ & \exp\left(\alpha_c + \bar{\boldsymbol{\beta}}_c^T \bar{\boldsymbol{x}}_i + \beta_x\right) \\ =& \frac{\sum_{i=1}^{K} I(k \neq c) \exp\left(\alpha_k + \boldsymbol{\beta}_k^T \boldsymbol{x}_i\right)}{\sum_{i=1}^{K} \exp\left(\alpha_k + \boldsymbol{\beta}_k^T \boldsymbol{x}_i\right)} \end{aligned}$$

Att derivatan är positiv för alla värden på β_x framgår tydligt i och med att både nämnaren och täljaren utgörs av operationer med multiplikation och addition av potenser av talet e, vilken alltid är positiv. Vidare ser vi att värdet på täljaren inte beror på β_x, medan nämnaren går mot oändligheten då β_x går mot oändligheten. Detta betyder att derivatan går mot 0 då β_x går mot oändligheten.

C Härledning av Wald-konfidensintervall för parametern i en binomialfördelad slumpvariabel

Vi ska härleda Wald-konfidensintervallet för parametern i en binomialfördelad slumpvariabel som ges av (12). Vi utgår från definitionen av Wald-konfidensintervall:

$$CI_{WALD} = \hat{\theta}_{ML} \pm \frac{\lambda_{0.025}}{\sqrt{I(\hat{\theta}_{ML})}}, \tag{13}$$

där $\hat{\theta}_{ML} = \frac{x}{N}$ Det vi måste hitta är $\hat{\theta}_{ML}$ och $I(\hat{\theta}_{ML})$. Vi börjar med $\hat{\theta}_{ML}$. Kärnan för likelihood-funktionen för ges av

$$\mathcal{L}(\theta : x) = \theta^x (1-\theta)^{N-x}.$$

Logaritmering ger log-likelihood-funktionen

$$\ell(\theta : x) = \log(\theta) \cdot x + \log(1-\theta) \cdot (N-x).$$

Derivering av log-likelihood-funktionen ger Score-funktionen

$$S(\theta) = \frac{x}{\theta} - \frac{N-x}{1-\theta}.$$

För att hitta $\hat{\theta}_{ML}$ sätter vi Score-funktionen lika med 0 och löser ut θ och tilldelar det värdet till $\hat{\theta}_{ML}$. Vi får även notera att det utlösta $\hat{\theta}_{ML}$ inte får vara 0 eller 1 eftersom Score-funktionen inte är definierad för dessa värden.

$$\begin{aligned} 0 &= S(\theta) \\ &= \frac{x}{\theta} - \frac{N-x}{1-\theta} \\ &= \frac{x - \theta x - \theta N + \theta x}{\theta(1-\theta)} \\ &= \frac{x - \theta N}{\theta(1-\theta)} \end{aligned}$$

Det enda sättet som det sista uttrycket i högerledet kan vara lika med 0 är om täljaren är lika med 0. Löser vi ut θ får vi

$$\hat{\theta}_{ML} = \frac{x}{N}.$$

Vi övergår nu till att hitta ett uttryck för fisherinformationen, $I(\theta)$. Denna definieras som den negativa andraderivatan av log-likelihood-funktionen. (Held and Bové, 2014 s.27), eller enklare i vårt fall som den negativa derivatan av Score-funktionen. Derivering av Score-funktionen ger att

$$\begin{aligned} I(\theta) &= -\frac{d}{d\theta}\left(\frac{x}{\theta} - \frac{N-x}{1-\theta}\right) \\ &= \frac{x}{\theta^2} + \frac{N-x}{(1-\theta)^2}. \end{aligned}$$

Sätter vi in $\hat{\theta}_{ML} = \frac{x}{N}$ i fisherinformationen får vi

$$\begin{aligned} I(\hat{\theta}_{ML}) &= \frac{x}{\left(\frac{x}{N}\right)^2} + \frac{N-x}{\left(1-\frac{x}{N}\right)^2} \\ &= \frac{N^2}{x} + \frac{N-x}{\frac{1}{N^2}(N-x)^2} \\ &= \frac{N^2}{x} + \frac{N^2}{N-x} \\ &= \frac{N^3}{x(N-x)} \\ &= N \cdot \frac{N}{x} \cdot \frac{N}{N-x} \\ &= N \cdot \left(\frac{x}{N}\right)^{-1} \cdot \left(\frac{N-x}{N}\right)^{-1} \\ &= N \cdot \hat{\theta}_{ML}^{-1} \cdot \left(1-\hat{\theta}_{ML}\right)^{-1} \end{aligned}$$

Nu behöver vi endast stoppa in de härledda uttrycken för $\hat{\theta}_{ML}$ och $I(\hat{\theta}_{ML})$ i (13). Detta ger oss Wald-konfidensintervallet

$$CI_{WALD} = \hat{\theta}_{ML} \pm \lambda_{0.025}\sqrt{\frac{1}{N}\hat{\theta}_{ML}(1-\hat{\theta}_{ML})},$$

vilket är exakt det vi skulle visa.

Källförteckning

[1] Agresti, A.: 2002, *Categorical Data Analysis*, 2nd edn, John Wiley & Sons, New Jersey.

[2] Aly, M.: 2005, Survey on multiclass classification methods. Teknisk Rapport. California Institute of Technology.

[3] Friedman, J., Hastie, T. and Tibshirani, R.: 2010, Regularization paths for generalized linear models via coordinate descent, *Journal of Statistical Software* **vol. 33**(1), s. 1–22.

[4] Hastie, T. and Qian, J.: 2014, *Glmnet Vignette.* `https://web.stanford.edu/~hastie/glmnet/glmnet_alpha.html` [2017-05-10].

[5] Hastie, T., Tibshirani, R. and Friedman, J.: 2016, *The Elements of Statistical Learning*, 2nd edn, Springer, New York.

[6] Held, L. and Bové, D. S.: 2014, *Applied Statistical Inference*, 1st edn, Springer, New York.

[7] James, G., Witten, D., Hastie, T. and Tibshirani, R.: 2015, *An Introduction to Statistical Learning*, 1st edn, Springer, New York.

[8] King, G. and Zeng, L.: 2001, Logistic regression in rare events data, *Political Analysis* **vol. 9**(2), s. 137–63.

[9] Larose, D. T. and Larose, C. D.: 2015, *Data Mining and Predictive Analytics*, 2nd edn, Wiley, New Jersey.

[10] Metz, C. E.: 1978, Basic principles of roc analysis, *Seminars in Nuclear Medicine* **vol 8**(4), s. 283–98.

[11] Ng, A.: n.d., *Advice for applying Machine Learning.* Föreläsningsanteckningar. `http://cs229.stanford.edu/materials/ML-advice.pdf` [2017-05-10].

[12] Olson, D. L. and Delen, D.: 2008, *Advanced Data Mining Techniques*, 1st edn, Springer, Berlin.

[13] Peduzzi, P., Concato, J., Kemper, E., Holford, T. R. and Feinstein, A. R.: 1996, A simulation study of the number of events per variable in logistic regression analysis, *Journal of Clinical Epidemiology* **vol. 49**(12), s. 1373–9.

[14] Perlich, C.: 2011, Learning curves in machine learning, I *Encyclopedia of Machine Learning*, Springer US, New York, pp. s. 577–580.

[15] Sundberg, R.: 2016, *Kompendium i Lineära Statistiska Modeller*, Stockholms Universitet, Stockholm.

[16] SCB (Statistiska Centralbyrån): n.d., *Standard för svensk näringsgrensindelning (SNI)*. `http://www.scb.se/sni` [2017-05-10].