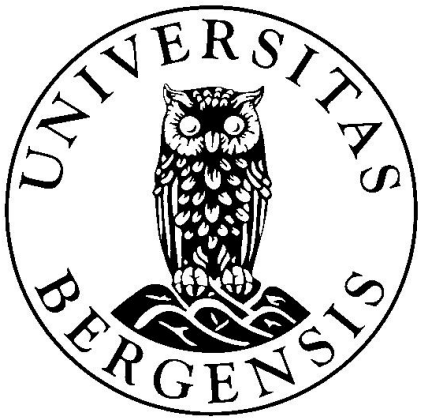
Statistiske modeller basert på skjulte Markovkjeder i kontinuerlig tid

Masteroppgave i statistikk - Finansteori og forsikringsmatematikk

Lisbet Lien Harjo



Matematisk institutt
Universitetet i Bergen

1. juni 2012

Takk

Jeg vil aller først gi en stor takk til Professor Ivar Heuch som har vært min veileder på denne oppgaven. Han har underveis gitt meg god inspirasjon, med mange gode tilbakemeldinger og innspill på oppgaven. Det har absolutt vært en svært lærerik prosess, som jeg kan ta med videre.

Jeg vil også takke mine gode medstudenter, venner og familie som har bidratt positivt til oppgaven og mitt liv som masterstudent generelt.

Lisbet Lien Harjo,
1. juni 2012

Innhold

Innledning

I denne oppgaven vil jeg ta for meg situasjoner der observasjoner over tid kan tenkes å være generert ved en Markovkjede, og studere innvirkningen av ulike kovariater på intensitetene, med både binære og tidavhengige kovariater. Jeg vil også se på situasjoner der tilstandene i Markovkjeden kan være observert feil, slik at man ikke virkelig observerer de underliggende tilstandene, men bare feilklassifiserte tilstander bestemt av en sannsynlighetsfordeling. Til slutt vil jeg også teste ut en aktuell pakke for programvaren `R` som er blitt utarbeidet for å kunne regne på denne typen problemstillinger.

Multitilstandsmodeller basert på Markovprosesser er en veletablert metode for å estimere overgangsratene mellom ulike sykdomsstadier. Dette grunnet av at mange sykdommer har en naturlig tolkning i form av en stadieprogresjon [10]. Både van den Hout og Matthews [20] og Jackson og Sharples [9] har anvendt dette i deres respektive artikler. Førstnevnte for å beskrive overganger mellom observasjoner over tid i en studie som omfatter slag og av sistnevnte for å forklare progresjonen av en kronisk sykdom. I mange slike studier vil diagnoser av sykdomsstadier noen ganger være feilaktige. Skjulte Markovmodeller er en forlengelse av Markovmodeller som gir en måte å redegjøre for potensielle feilklassifiseringer grunnet målingsprosedyren [1]. Det meste av teorien om skjulte Markovmodeller ble i hovedsak utviklet for tidsdiskrete modeller, der den har blitt mye brukt i områder som talegjenkjenning [11] og analyse av biologiske sekvensdata [6]. I tekniske og biologiske sekvenseringsapplikasjoner, utvikler Markovprosessen seg som regel over et likt fordelt, diskret tidsrom [8]. Skjulte Markovmodeller var mindre brukt innenfor medisin, der tidskontinuerlige prosesser ofte er mer passende. En sykdomsprosess utvikler seg i kontinuerlig tid, og pasienter er ofte overvåket ved uregelmessige og forskjellige intervaller.

Jeg har i oppgaven fokusert på det som er blitt utviklet for tidskontinuerlige prosesser. Jeg studerte først en artikkel fra 2009 av van den Hout m. fl. [19], der de har brukt en tidskontinuerlig, diskre tilstand skjult Markovmodell. For å få en større forståelse på hvordan denne modellen var bygget opp, studerte jeg videre en annen artikkel fra 2003 av Jackson m. fl. [10], der den generelle skjulte Markovmodellen ble satt opp for tidskontinuerlige prosesser. Forfatterene har der generalisert modellen til å kunne bli tilpasset ulike overgangs- og feilklassifiseringsmuligheter. De presenterer også en ny pakke for programvaren `R` som de har kalt *msm*. Denne skal kunne tilpasses tidskontinuerlige og skjulte Markovmodeller, og Jackson [8] har i senere tid skrevet en manual som omhandler hvilke funksjoner som ligger i pakken og vist til hvordan den fungerer.

Oppgaven er bygget opp i hovedsaklig to deler. Først en teoretisk del, der jeg i kapittel 1-2 beskriver de ulike modellene som er blitt brukt i artiklene og i kapittel 3-5 setter opp mitt eget eksempel. Der setter jeg først opp uttrykkene for overgangssannsynlighetene og likelihooden, før jeg videre inkluderer muligheten for feilklassifisering. Derpå følger det en praktisk del i kapittel 6-8, der jeg i detalj går igjennom msm-pakken, og prøver å anvende pakken på simulerte datasett. Oppgaven avsluttes så med en konklusjon og videre arbeid i kapittel 9.

Kapittel 1

Markovmodeller

1.1 Markovkjeder

En *Markovprosess* $\{X_t\ ; t \in T\}$ er en stokastisk prosess med den egenskapen at, gitt en verdi av X_t, er verdiene av X_s for $s > t$ ikke påvirket av verdiene av X_u for $u < t$ [17], kap. 3. Med andre ord, hvis den nåværende tilstanden i prosessen er kjent, vil ikke sannsynligheten for en bestemt hendelse i framtiden bli påvirket av at en har tilleggskunnskaper om det som har skjedd tidligere i prosessen, altså før den nåværende tilstanden. En *tidsdiskret* Markovkjede er en Markovprosess der tilstandsrommet er en endelig eller tellbar mengde, og der (tids-)indeksmengden er $T = (0, 1, 2, \dots)$. Formelt er Markovegenskapen:

$$\Pr(X_{n+1} = j | X_0 = i_0, \dots, X_{n-1} = i_{n-1}, X_n = i) = \Pr(X_{n+1} = j | X_n = i)$$

for alle tidspunkter n og alle tilstandene $i_0, ..., i_{n-1}, i, j$.

$P_{ij} = \Pr(X_{n+1} = j | X_n = i)$ er den betingede sannsynligheten for at det vil skje en overgang fra tilstand i til tilstand j på et forsøk. $\mathbf{P} = \|P_{ij}\|$ blir referert som *Markovmatrisen* eller *overgangssannsynlighetsmatrisen* av prosessen. Den i'te rekken av $\mathbf{P}$, for $i = 0, 1, \dots$, er sannsynlighetsfordelingen av verdiene til X_{n+1} under betingelsen at $X_n = i$. Tallene P_{ij} tilfredsstiller betingelsene $P_{ij} \geq 0$ for $i, j = 0, 1, 2, \dots$, og $\sum_{j=0}^{\infty} P_{ij} = 1$ for $i = 0, 1, 2, \dots$. Den siste betingelsen forteller at det ved hvert steg skjer en overgang selv om tilstanden forblir den samme.

En Markovmodell er en stokastisk modell som antar Markovegenskapen. Med denne antakelsen er det mulig å gjøre resonnementer og beregninger med modellen som ellers ville vært uløselige. Markovmodeller er en praktisk og nyttig metode for å estimere overgangsrater mellom nivåer av en kategorisk responsvariabel, slik som et sykdomsstadie, som endres over tid [18].

1.2 Tidskontinuerlige Markovkjeder

En tidkontinuerlig Markovkjede X_t $(t > 0)$ er en Markovprosess på tilstandene $0, 1, 2, \dots$ [17], kap. 6. Antar at overgangssannsynlighetene er stasjonære, som betyr

at

$$P_{ij}(t) = \Pr\{X_{t+s} = j | X_s = i\}.$$

Istedenfor at det ved hvert steg skjer en overgang fra en tilstand til en annen (muligens den samme), som ved en tidsdiskret Markovkjede, vil en *tidskontinuerlig* Markovkjede befinne seg i dens nåværende tilstand for en tilfeldig, ofte eksponensielt fordelt tid for så å hoppe videre til en annnen tilstand. Generelt skal intensitetene for en gitt tilstand summeres til null, slik at diagonalelementene blir

$$q_i = -\sum_{j=0, j\neq i} q_{ij}.$$

Disse intensitetene gir en infinitesimal beskrivelse av prosessen (i et ekstremt lite tidsintervall dt) med følgende sannsynligheter

$$\begin{aligned} \Pr(X_{t+dt} = j | X_t = i) &= q_{ij}dt + o(dt) \qquad i \neq j, \\ \Pr(X_{t+dt} = i | X_t = i) &= 1 - q_i dt + o(dt), \end{aligned}$$

der $o(dt)$ representerer en ubetydelig liten reststørrelse, som ved å dividere leddet med dt, vil gå raskere til null enn dt går mot null. Intensiteten q_{ij}, $i \neq j$, måler hvor raskt en overgang fra tilstand i til tilstand j skjer og er det (i, j)-elementet i overgangsintensitetsmatrisen $\mathbf{Q}$, også kalt *intensitetsmatrisen*.

En tidsavhengig Markovprosess er en Markovprosess som ovenfor, men med intensitetene som en funksjon av tid, skrevet som $q_{ij}(t)$.

1.3 Skjulte Markovmodeller

En skjult Markovmodell er en Markovkjede observert i støy [2]. Modellen består av en Markovkjede $\{X_t\}_{t\geq 0}$ som ofte antas å ta verdier i en endelig mengde. Markovkjeden er nå *skjult*, som betyr at den ikke er observerbar. Det som er tilgjengelig for observatøren er en annen stokastisk prosess $\{Y_t\}_{t\geq 0}$ som er knyttet til Markovkjeden i at X_t styrer fordelingen av den tilsvarende Y_t. All statistisk inferens, også på selve Markovkjeden, må skje med hensyn på bare $\{Y_t\}$, siden $\{X_t\}$ ikke er observert.

Cappé m. fl. [2] har gitt følgende definisjon: En skjult Markovmodell er en bivariat tidsdiskre prosess $\{X_t, Y_t\}_{t\geq 0}$, der $\{X_t\}$ er en Markovkjede og betinget på $\{X_t\}$, er $\{Y_t\}$ en sekvens av uavhengige tilfeldige variabler slik at den betingede fordelingen av Y_t bare avhenger av X_t.

Skjulte Markovmodeller er anvendt i et bredt spekter av områder, som inkluderer talegjenkjenning, økonometri, bildeanalyse og bioinformatikk [15].

Kapittel 2

Modellene brukt i artiklene

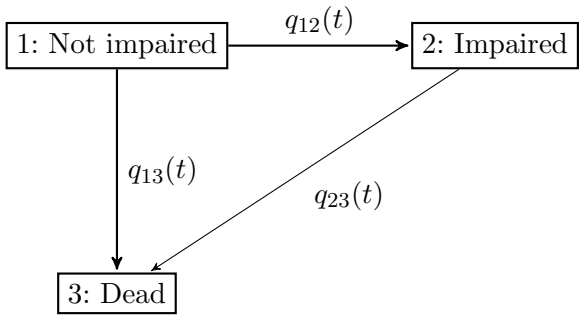
2.1 Modellene

Jeg vil nå beskrive de ulike modellene som er blitt brukt i artiklene til van den Hout m. fl. [19] og Jackson m. fl. [10]. Van den Hout m. fl. bruker en skjult Markovmodell for å beskrive den kognitive nedgangen i den eldre populasjonen, der en observert forbedring av kognitiv evne blir modellert som en feilklassifisering. Videre har de brukt maksimum likelihood for å estimere overgangsintensitetene mellom normal kognitiv tilstand, svekket kognitiv tilstand og dødstilstanden. Jackson m. fl. presenterer en generell skjult Markovmodell for å beskrive sykdomsstadier, som samtidig estimerer overgangsratene og sannsynlighetene for feilklassifisering av stadiene. De har illustrert modellen på data med bakgrunn fra en studie av aortaaneurisme skanning, der skanningsmålingene er utsatt for feil.

2.2 Beskrivelse av modellene til van den Hout m. fl.

Mange populasjonsstudier har observasjoner som går over tid og dødsinformasjon som kan brukes til å estimere overganger mellom friske og ikke-friske tilstander før død [19]. I studien som van den Hout m. fl. betrakter, er observasjoner av kognitiv evne vesentlige som en viktig prediktor av den eldre befolkningens helse. Generelt er *trajektoriene*, bevegelsesretningene mellom tilstandene, av kognitiv evne antatt å være statiske eller nedovergående med økende alder, slik at en observert forbedring skal behandles som en feilklassifisering.

Det er i artikkelen blitt brukt en tidskontinuerlig, diskre tilstand skjult Markovmodell for å estimere total levealder og kognitivt svekket levealder. Svekket levealder er her definert som forventet gjenværende levetid som kognitivt svekket. Siden trajektoriene av kognitiv evne ikke kan gå oppover, at den ikke kan bli bedre, velger de å anvende en skjult Markovmodell som tillater feilklassifiseringer av kognitive tilstander. Dette betyr at for en oppovergående trajektorie, fra en svekket tilstand til en ikke-svekket tilstand, som enten kommer av at en tilstand er direkte feilklassifisert, eller den kan være forårsaket av en tidligere feilaktig observert nedovergående trajektorie. Kan heller ikke utelukke feilklassifisering i de tilfellene uten observert forbedring.



Figur 2.1: Tretilstandsmodellen brukt av van den Hout m. fl.

En tidskontinuerlig skjult Markovmodell er en multitilstandsmodell der Markov-antakelsen er formulert med hensyn på de latente tilstandene [19]. Modellen for sykdomsprogresjon er et eksempel på en multitilstandsmodell der individer stadig går til et mer alvorligere stadie av sykdommen, for til slutt å havne i en absorberende tilstand, som ofte vil være døden. Sykdom-og-dødsmodellen er en tre-tilstandsmodell for sykdomsprogresjon med to transiente tilstander (frisk og syk) og en absorberende tilstand (død). Figur 2.1 viser sykdom-og-dødsmodellen som er blitt brukt i denne artikkelen.

De utvider den skjulte Markovmodellen i Satten og Longini [16] ved å inkludere en logistisk regresjonsmodell for fordelingen av initialtilstanden, for å kunne estimere levetidene.

2.2.1 Skjult Markovmodell

Van den Hout m. fl. [19] innfører notasjonene for tre-tilstandsmodellen og presenterer en tilpasning ved å inkludere logistisk regresjon for den latente (uobserverbare) initialtilstandsfordelingen. Lar tid t være tiden som har gått siden starten av studien. For $t \geq 0$, la $X_t \in \{1, 2, 3\}$ være de sanne tilstandene for et individ, og $X_t^* \in \{1, 2, 3\}$ være de observerte tilstandene. Klassifiseringssannsynlighetene $c_{rs} = \Pr(X_t^* = s | X_t = r)$ gir matrisen

$$\mathbf{C} = \begin{bmatrix} c_{11} & c_{12} & c_{13} \\ c_{21} & c_{22} & c_{23} \\ c_{31} & c_{32} & c_{33} \end{bmatrix} = \begin{bmatrix} 1 - c_{12} & c_{12} & 0 \\ c_{21} & 1 - c_{21} & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

De får bare to parametere som skal estimeres siden tilstand 3 (døden) er målt uten feilklassifisering, og siden rekkene skal summeres til 1. En overgang fra tilstand r til tilstand s der $r \neq s$, skjer med en intensitet $q_{rs}(t)$, der $q_{rs}(t) > 0$ for $(r, s) \in \{(1, 2), (1, 3), (2, 3)\}$, og $q_{rs}(t) = 0$ for $(r, s) \in \{(2, 1), (3, 1), (3, 2)\}$. Intensiteten $q_{rs}(t)$ representerer den momentane risikoen for å hoppe fra tilstand r til tilstand s ved tid t. De lar intensitetene avhenge av kovariater $\mathbf{z}(t)$ via en regresjonsmodell

$$q_{rs}(t) = \exp\{\beta^T \mathbf{z}(t)\},$$

der første leddet i $\mathbf{z}(t)$ er antatt til å være lik 1. Kovariatene kan også være tidsavhengige. Disse intensitetene former intensitetsmatrisen $\mathbf{Q}(t)$, som er gitt ved

$$\mathbf{Q}(t) = \begin{bmatrix} -(q_{12}(t) + q_{13}(t)) & q_{12}(t) & q_{13}(t) \\ 0 & -q_{23}(t) & q_{23}(t) \\ 0 & 0 & 0 \end{bmatrix},$$

der en generell egenskap for intensitetsmatriser er at rekkene skal summeres til 0. Selv om det ikke nevnes i artikkelen er det også fullt mulig å legge til kovariater på klassifiseringssannsynlighetene c_{rs}. Noen av c_{rs}-ene kan være bestemt slik at de reflekterer kunnskap om diagnoseprosessen, som jeg vil komme tilbake til i gjennomgangen av artikkelen til Jackson m. fl. [10]. Individene er målt over tid, der både tidspunktene og tidsintervallene kan variere mellom og innen individer. De håndtrerer denne tidsavhengigheten ved å bruke en stykkevis konstant tilnærming, der intensitetene ikke endres innenfor det intervallet som er definert ut fra to påfølgende individuelle observasjonstider. Intensitetsmatrisen innenfor et intervall bestemmes av de kovariatverdiene som er gjeldendes ved inngangen av det observerte tidsintervallet. For en gitt tidsavhengig kovariat, som for eksempel alder, vil intensitetene endre seg fra det ene intervallet til det neste, men ikke innenfor det bestemte tidsintervallet.

Videre i artikkelen sier van den Hout m. fl. at hvis intensitetene er konstante i et tidsintervall $(t, u]$ vil den tilhørende overgangssannsynlighetsmatrisen være $\mathbf{P}(t, u) = \exp\{(u-t)\mathbf{Q}(t)\}$, med elementene $p_{rs} = \Pr\{X_u = s | X_t = r, \mathbf{z}(t)\}$, for $r, s \in \{1, 2, 3\}$. Dette er et resultat som kommer av at de arbeider med en tidskontinuerlig Markovkjede. Taylor og Karlin [17], kap. 6, har vist hvordan de med Chapman-Kolmogorovs relasjonen

$$P_{ik}(s + t) = \sum_{j=0}^{N} P_{ij}(s) P_{jk}(t) \qquad \text{for } t, s \geq 0,$$

og med

$$\lim_{t \to 0+} P_{ij}(t) = \begin{cases} 1 & i = j, \\ 0 & i \neq j, \end{cases}$$

kan oppnå uttrykket

$$\mathbf{P}(t) = e^{\mathbf{A}t} = \mathbf{I} + \sum_{n=1}^{\infty} \frac{\mathbf{A}^n t^n}{n!},$$

der $\mathbf{A}$ er intensitetsmatrisen.

2.2.2 Likelihood

Antar at et individ har observasjoner ved tidene $t_1, \ldots, t_m$. Ved å bruke loven om total sannsynlighet blir bidraget av dette individet til likelihooden

$$L = \Pr(X^*_{t_1}, \ldots, X^*_{t_m}) = \sum \Pr(X^*_{t_1}, \ldots, X^*_{t_m} | X_{t_1}, \ldots, X_{t_m}) \Pr(X_{t_1}, \ldots, X_{t_m})$$

der summen er over alle mulige veier av latente (uobserverte) tilstander $X_{t_1}, \ldots, X_{t_m}$, og avhengigheten av kovariatene er ignorert i notasjonen. De bruker så Markovegenskapen på $\Pr(X_{t_1}, \ldots, X_{t_m})$ som gir

$$\begin{aligned}
\Pr(X_{t_1}, \ldots, X_{t_m}) &= \Pr(X_{t_m}|X_{t_1}, \ldots, X_{t_{m-1}}) \Pr(X_{t_1}, \ldots, X_{t_{m-1}}) \\
&= \Pr(X_{t_m}|X_{t_{m-1}}) \Pr(X_{t_1}, \ldots, X_{t_{m-1}}) \\
&= \Pr(X_{t_m}|X_{t_{m-1}}) \Pr(X_{t_{m-1}}|X_{t_1}, \ldots, X_{t_{m-2}}) \Pr(X_{t_1}, \ldots, X_{t_{m-2}}) \\
&= \Pr(X_{t_m}|X_{t_{m-1}}) \Pr(X_{t_{m-1}}|X_{t_{m-2}}) \Pr(X_{t_1}, \ldots, X_{t_{m-2}}) \\
&\vdots \\
&= \Pr(X_{t_m}|X_{t_{m-1}}) \Pr(X_{t_{m-1}}|X_{t_{m-2}}) \ldots \Pr(X_{t_2}|X_{t_1}) \Pr(X_{t_1}),
\end{aligned}$$

og den betingede uavhengigheten av de observerte tilstandene gitt de latente tilstandene som gir

$$\begin{aligned}
L = &\sum_{X_{t_1}} \Pr(X^*_{t_1}|X_{t_1}) \Pr(X_{t_1}) \sum_{X_{t_2}} \Pr(X^*_{t_2}|X_{t_2}) \Pr(X_{t_2}|X_{t_1}) \\
&\cdots \sum_{X_{t_m}} \Pr(X^*_{t_m}|X_{t_m}) \Pr(X_{t_m}|X_{t_{m-1}}),
\end{aligned} \tag{2.1}$$

der $\Pr(X^*_t|X_t)$ er klassifikasjonssannsynlighetene hentet fra $\mathbf{C}$ og $\Pr(X_{t_{j+1}}|X_{t_j})$, $j = 1, \ldots, m-1$ er overgangssannsynlighetene fra $\mathbf{P}(t_j, t_{j+1})$. Van den Hout m. fl. [19] foreslår at man kan estimere fordelingen av den første latente tilstanden, $\Pr(X_{t_1})$, ved å bruke en logistisk regresjonsmodell

$$\Pr(X_{t_1} = 1) = \frac{\exp\{\gamma^T \mathbf{z}(t_1)\}}{1 + \exp\{\gamma^T \mathbf{z}(t_1)\}},$$

der $\Pr(X_{t_1} = 2) = 1 - \Pr(X_{t_1} = 1)$. I tillegg har de at $\Pr(X_{t_1} = 3) = 0$. Legger merke til at γ er regresjonskoeffisientene brukt ved grunnlinjen (linjen brukt som utgangspunkt) og er ikke de samme som regresjonskoeffisientene β, brukt i intensitetsmatrisen $\mathbf{Q}$.

De velger å bruke matriser til å beskrive likelihooden. Lar $\mathbf{f}$ være en 1×3 rekkevektor for den første summen i 2.1 med r'te elementet $\Pr(X^*_{t_1}|X_{t_1} = r) \Pr(X_{t_1} = r)$ for $r \in \{1, 2, 3\}$. Lar $\mathbf{T}_j$ for $j = 2, \ldots, m-1$ være en 3×3 matrise, der hver $\mathbf{T}_j$ representerer en summasjon i 2.1 med (r, s)-posisjonen

$$\Pr(X^*_{t_j}|X_{t_j} = s) \Pr(X_{t_j} = s|X_{t_{j-1}} = r).$$

For den siste summen i 2.1 vil 3×3 matrisen, $\mathbf{T}_m$, være forskjellig ut fra om individet er død eller levende. I tilfellet med død, $X^*_{t_m} = 3$, er (r, s)-posisjonen av $\mathbf{T}_m$ lik

$$\Pr(X_{t_m} = s|X_{t_{m-1}} = r) q_{s3}(t_m),$$

der de antar en ukjent latent tilstand s ved tid t_m og så en umiddelbar død.

I tilfellet med sensurering, er det kjent at individet fremdeles er i live, men at tilstanden er ukjent. I dette tilfellet er (r, s)-posisjonen av $\mathbf{T}_m$

$$\Pr(X_{t_m} = s | X_{t_{m-1}} = r)\mathbf{1}_{[s\in\{1,2\}]},$$

der $\mathbf{1}_{[A]}$ er lik 1 dersom A er sann og 0 ellers. Med å bruke matrisenotasjon fremfor summer, får de at den enkeltes bidrag til likelihooden vil bli

$$L = \mathbf{f}\mathbf{T}_2\mathbf{T}_3 \times ... \times \mathbf{T}_m\mathbf{1}$$

der $\mathbf{1} = (1, 1, 1)^T$. Videre estimerer de parameterne av den samlede modellen, som her er parameterne β, c_{12}, c_{21} og γ ved å maksimere log-likelihooden.

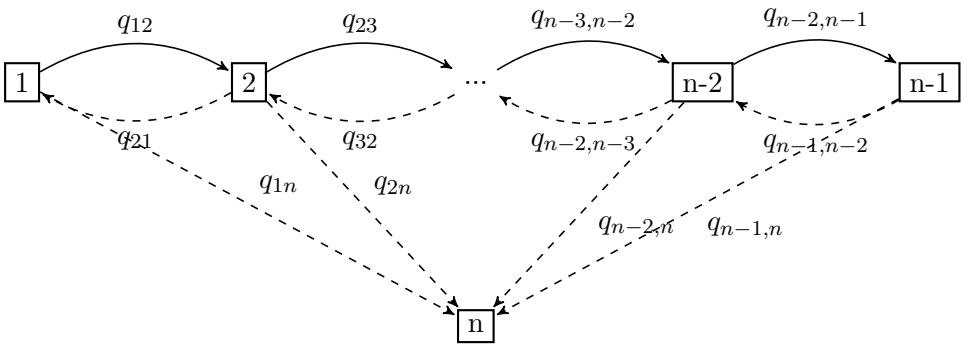
2.3 Beskrivelse av modellene til Jackson m. fl.

Artikkelen [10] presenterer et generelt rammeverk for stadie Markovmodeller med feilklassifisering, der alle former av overgangsmatriser er tillatt, med feilklassifiseringer mellom ethvert par av tilstander. De bemerker at en sykdom kanskje er mer mottakelig for behandling dersom den oppdages på et tidlig stadie, og at systematisk skanning av en populasjon kan være en effektiv måte å redusere dødligheten av en sykdom på. For å etablere en tilstrekkelig skanningspolise for sykdommen kreves det at man har kunnskap om dens naturlige historie. Hvilke individer som skal skannes og tiden for skanningen, bør bli valgt i henhold til risikoen for et utbrudd av sykdommen. Intervallene mellom suksessive skanninger bør også velges i samsvar med risikoen for progresjon.

2.3.1 Multitilstands Markovmodeller

Tidskontinuerlige multitilstands Markovmodeller er mye brukt til å modellere sykdomsutviklinger. I figur 2.2 viser Jackson m. fl. [10] en ofte brukt modell som representerer en serie med suksessivt mer alvorligere stadier av en sykdom og en absorberende tilstand som ofte er døden. En pasient kan i denne modellen enten bli dårligere eller bedre, eller død ved en av tilstandene. For et antall individer i blir det ved en vilkårlig tid t gjort observasjoner av stadiet $S_i(t)$ De bruker en homogen tidskontinuerlig Markovprosess til å modellere sykdomsstadiene. Modellen i figur 2.2 kan bli beskrevet av overgangsintensitetsmatrisen $\mathbf{Q}$

$$\mathbf{Q} = \begin{bmatrix} q_{11} & q_{12} & 0 & 0 & \dots & q_{1n} \\ q_{21} & q_{22} & q_{23} & 0 & \dots & q_{2n} \\ 0 & q_{32} & q_{33} & q_{34} & \ddots & q_{3n} \\ 0 & 0 & q_{43} & q_{44} & \ddots & q_{4n} \\ \vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & \dots & 0 \end{bmatrix},$$



Figur 2.2: Den generelle modellen for sykdomsprogresjon

der rekkene summeres til 0, slik at diagonalelementene er $q_{rr} = -\sum_{s \neq r} q_{rs}$. Maksimum likelihood estimatene for denne klassen av modeller kan beregnes fra overgangssannsynlighetsmatrisen $\mathbf{P}(t)$, med (r, s)-elementer

$$p_{rs}(t) = \Pr(S_i(t+u) = s | S_i(u) = r).$$

I likhet med tidligere, hos van den Hout m. fl. [19] i seksjon 2.2.1, avhenger denne av intensitetene i $\mathbf{Q}$ gjennom Kolmogorov forholdet $\mathbf{P}(t) = \exp(t\mathbf{Q})$. Sykdommen som studeres antas i noen tilfeller for å være irreversibel. Overgangsintensitetene som tilsvarer bedringer av sykdommen settes da til å være lik null. Skanningsprosedyren av sykdommen kan også være forbeholdt feil. Markovprosessen $S_i(t)$ er da ikke direkte observert, men gjennom observasjonene $O_i(t)$.

2.3.2 Skjult Markovmodell

Jackson m. fl. [10] definerer videre den generelle modellen for sykdomsprogresjon og diagnosefeil. De lar i indikere I individer, j indikere de m_i observasjonstidene for hvert individ i, og lar n være antall tilstander. Antar at $S_{ij} = S_i(t_{ij})$ representerer den sanne underliggende tilstanden til individ i ved tid t_{ij}, og at O_{ij} representerer den tilsvarende observerte tilstanden.

For individ i antas S_{ij} for å være realiseringer fra en multitilstandsprosess $S_i(t)$ med intensitetsmatrisen $\mathbf{Q}$. Elementet på (r, s)-posisjonen i $\mathbf{Q}$ representerer hasardraten for progresjon til stadie s, betinget på å være i stadie r:

$$q_{rs} = q_{rs}(t, \mathcal{F}_t) = \lim_{\delta t \to 0} \frac{\Pr\{S_i(t+\delta t) = s | S_i(t) = r, \mathcal{F}_t\}}{\delta t},$$

der $\mathcal{F}_t$ er observasjonshistorien av prosessen opp til tiden forut t. Hvis prosessen $S_i(t)$ er antatt til å være tidshomogen og Markov er $q_{rs}(t, \mathcal{F}_t)$ uavhengig av t og $\mathcal{F}_t$. O_{ij} genereres betinget på S_{ij} i henhold til en feilklassifiseringsmatrise $\mathbf{E}$. Dette er en $n \times n$ matrise, der (r, s)-posisjonen er

$$e_{rs} = \Pr(O(t_{ij}) = s | S(t_{ij}) = r),$$

som de først antar for å være uavhengig av tiden t. Analogt til $\mathbf{Q}$ kan noen av e_{rs}-ene være bestemt til å reflektere kjennskap til diagnoseprosessen. F.eks. kan sannsynligheten for feilklassifisering kanskje være ubetydelig liten for tilstander av sykdommen som ikke er nabotilstander. Dette er et eksempel på en skjult Markovmodell. Progresjonen gjennom underliggende tilstander og observasjonsprosessen av de underliggende tilstandene er dekket av separate modeller.

2.3.3 Kovariater

Jackson m. fl. [10] forteller videre at forklaringsvariabler kan gjennom generalisert regresjon bli inkludert på hvert nivå av modellen. En proporsjonal hasardmodell kan bli brukt til å relatere overgangsintensitetene q_{rs} ved tid t til kovariater $z(t)$ ved den tiden

$$q_{rs}(z_{ij}) = q_{rs}^{(0)} \exp(\beta_{rs}^T z_{ij}).$$

Den nye $\mathbf{Q}$ er da brukt til å beregne likelihooder. Er kovariatene z_t tidsavhengige vil Markovkjedens overgangssannsynlighet $p_{rs}(t_2 - t_1)$ mellom tilstandene r og s fra tid t_1 til tid t_2 bli erstattet med

$$p_{rs}\{t_2 - t_1, z(t_1)\},$$

selv om det krever at verdien av kovariaten skal være kjent ved hver observasjonstid t_1. Noen ganger vil kovariater være observert ved forskjellige tider til hovedresponsen, som ved rekurrente sykdomshendelser eller andre biologiske markører. Kovariaten kan i noen av disse situasjonene antas for å være en stegfunksjon som forblir konstant mellom dens observasjonstider. Selv om verdien av kovariaten er kjent til enhver tid, som alder til pasienten, vil overgangsintensitetene mellom tidene t_1 og t_2 fremdeles antas å bare avhenge av kovariatverdien som er gjeldende ved tid t_1.

Tilsvarende for å undersøke forklaringsvariablene $w(t)$ ved sannsynligheten e_{rs} for feilklassifisering, kan det for hvert par av stadiene r og s bli brukt en logistisk modell:

$$\log \frac{e_{rs}(t)}{1 - e_{rs}} = \gamma_{rs}^T w(t).$$

2.3.4 Maksimum likelihood estimering

Macdonald and Zucchini [21] har beskrevet en direkte metode for å beregne likelihooder i diskre og kontinuerlig tid basert på matriseprodukter. Satten og Longini [16] brukte denne metoden til å beregne likelihooden for en skjult Markovmodell i kontinuerlig tid med observasjoner av en kontinuerlig markør generert betinget på underliggende diskrete tilstander. Jackson m. fl. [10] illustrerer her matriseproduktmetoden for feilklassifiseringsmodellen. Den kan generaliseres til en hvilken som helst form for data som er generert betinget på tilstandene av en skjult Markovprosess.

Bidraget til likelihooden fra individ i er

$$\begin{aligned} L &= \Pr(O_{i1}, \ldots, O_{im_i}) \\ &= \sum \Pr(O_{i1}, \ldots, O_{im_i} | S_{i1}, \ldots, S_{im_i}) \Pr(S_{i1}, \ldots, S_{im_i}), \end{aligned} \tag{2.2}$$

der summen er tatt over alle mulige veier av underliggende tilstander $S_{i1}, \ldots, S_{im_i}$. I likhet med de antagelsene som ble gjort for likelihooden i seksjon 2.2.2. kan den totale summen i likning 2.2 deles inn i summer over hver underliggende stadie. Summen er akkumulert over den ukjente første tilstanden, den ukjente andre tilstanden og så videre opptil den ukjente endelige tilstanden:

$$L_i = \sum_{S_{i1}} \Pr(O_{i1}|S_{i1}) \Pr(S_{i1}) \sum_{S_{i2}} \Pr(O_{i2}|S_{i2}) \Pr(S_{i2}|S_{i1}) \sum_{S_{i3}} \Pr(O_{i3}|S_{i3}) \Pr(S_{i3}|S_{i2}) \cdots \sum_{S_{im_i}} \Pr(O_{im_i}|S_{im_i}) \Pr(S_{im_i}|S_{im_{i-1}}), \tag{2.3}$$

der $\Pr(O_{ij}|S_{ij})$ er feilklassifiseringssannsynligheten $e_{S_{ij}O_{ij}}$, mens $\Pr(S_{i,j+1}|S_{ij})$ er $(S_{ij}, S_{i,j+1})$-posisjonen av overgangsmatrisen $\mathbf{P}(t)$ evaluert ved $t = t_{i,j+1} - t_{ij}$. Lar $\mathbf{f}$ være vektoren av de første stadiene okkupasjonssannsynlighetene $\Pr(S_{i1})$, og lar $\mathbf{1}$ være en vektor bestående bare av 1-ere. For $j = 2, \ldots, m_i$, la T_{ij} være en $n \times n$ matrise med (r, s)-posisjonen

$$e_{sO_{ij}} p_{rs}(t_{ij} - t_{i,j-1}).$$

Ved å heller skrive $e_{sO_{ij}}$ som $\Pr(O_{ij}|s)$, ser jeg at uttrykket for T_{ij} ligner på det som ble gitt hos van den Hout m. fl. [19] for T_j. Bidraget til likelihooden for individ i kan da bli skrevet som et produkt av matriser

$$L_i = \mathbf{f} T_{i2} T_{i3}, \ldots, T_{im_i} \mathbf{1},$$

som tilsvarer likelihooden fra artikkelen til van den Hout m. fl.

2.3.5 Eksempel

Jackson m. fl. [10] anvender den generelle modellen på et datasett som er hentet fra en studie av aortaaneurismeskanning. Veksten av aneurysmen antas å være en irreversibel prosess, slik at en mulig intensitetsmatrise kan være

$$\mathbf{Q} = \begin{bmatrix} -q_{12} & q_{12} & 0 & 0 \\ 0 & -q_{23} & q_{23} & 0 \\ 0 & 0 & -q_{34} & q_{34} \\ 0 & 0 & 0 & 0 \end{bmatrix},$$

som indikerer en progresjon mellom to påfølgende tilstander, med ingen hoppmuligheter fra stadie 4. Tilstandene tilsvarer i denne studien en økende aortadiameter. Målingene av aortadiameteren er utsatt for feil, der overganger fra høyere til lavere tilstander ofte er blitt observert. Det å tilpasse en multitilstandsmodell som bare tillater progresjon, en økende utvikling, med en intensitetsmatrise $\mathbf{Q}$ og ingen feilklassifiseringer, krever at de reverse overgangene må fjernes eller blir glattet over på en eller annen måte. For den skjulte Markovmodellen, anta at de uobserverte sanne tilstandene følger en Markovprosess med overgangsmatrisen $\mathbf{Q}$, og at de observerte tilstandene er generert fra de latente tilstandene gjennom feilklassifiseringssannsynlighetsmatrisen

$$\mathbf{E} = \begin{bmatrix} 1 - e_{12} & e_{12} & 0 & 0 \\ e_{21} & 1 - e_{21} - e_{23} & e_{23} & 0 \\ 0 & e_{32} & 1 - e_{32} - e_{34} & e_{34} \\ 0 & 0 & e_{43} & 1 - e_{43} \end{bmatrix}.$$

2.4 Oppsummering

Den generelle Markovmodellen for feilklassifisering ble utarbeidet av Jackson m. fl. [10], der alle former for overgangsintensitetsmatriser er tillatt, og med feilklassifiseringer mellom hvilke som helst tilstander. De utviklet også en pakke for programvaren `R` som er gjort lett tilgjengelig via internettsiden http://cran.r-project.org/. Van den Hout m. fl. [19] har i sin artikkel fra 2009 referert til Jackson m. fl., der de anvender den generelle modellen, men med bruk av litt andre notasjoner. De har utvidet den skjulte Markovmodellen ved å inkludere en logistisk regresjonsmodell på fordelingene av de første latente tilstandene.

Jackson m. fl. har vist hvordan modellen fungerer for økende sykdomsstadier, mens van den Hout m. fl. har illustrert den på en sykdom- og dødsmodell med tre tilstander. Det er i artiklene blitt brukt litt forskjellige notasjoner. Jackson m. fl. har blant annet brukt notasjonene $S_i(t)$ og $O_i(t)$ for de sanne og observerte tilstandene, der van den Hout m. fl. har brukt henholdsvis $X(t)$ og $X^*(t)$.

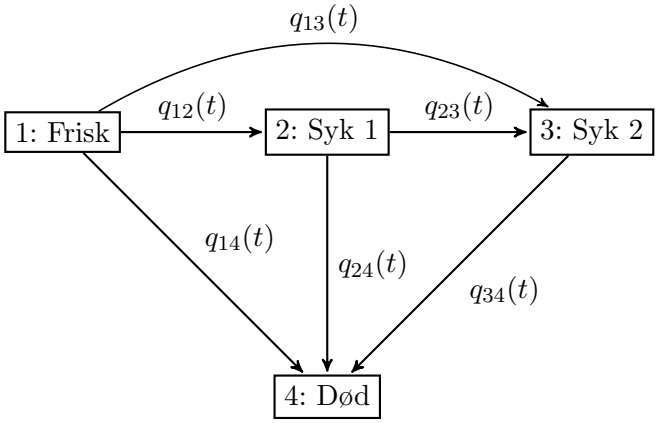
De har valgt å sette opp kovariatene litt forskjellig, der de bruker ulike notasjoner for å skrive opp det første leddet i intensitetene. Jackson m. fl. har leddet $q_{rs}^{(0)}$ som viser til intensiteten uten påvirkning av kovariater, mens Van Den Hout har valgt å inkludere et ledd til i $\mathbf{z}(\mathrm{t})$, 1, slik at de kan skrive hele uttrykket for intensitetene ved hjelp av vektorer. Begge uttrykkene betyr her det samme.

Kapittel 3

Markovmodell med fire tilstander

3.1 Eget spesialtilfelle

Etter å ha studert van den Hout m. fl. [19] sin tre-tilstandsmodell, kan det være interessant å utvide denne modellen med en ekstra tilstand, se figur 3.1. En sykdom har ofte en gradvis sykdomsutvikling og en mulig utvidelse kan dermed være å dele sykdomstilstanden inn i to tilstander. Lar den ene tilstanden være for de som befinner seg i et tidlig eller mindre alvorlig stadie av en sykdom, og lar den andre tilstanden være for de som befinner seg i et mer alvorlig sykdomsstadie. Individene som er med i studien er målt over tid, der både tidspunktene og tidsintervallene kan variere fra person til person. Velger at for denne studien kan det være mulig for et individ å bevege seg fra tilstand 1 til tilstand 3 på to forskjellige måter. Individet kan enten ha vært innom tilstand 2 i et lite tidsintervall mellom observasjonstidene før den hopper videre til tilstand 3, eller ha hoppet direkte fra tilstand 1 til tilstand 3.



Figur 3.1: Fire-tilstandsmodellen.

Lar t angi tiden som er gått siden oppstarten av studien, og for $t \geq 0$ lar jeg

$X_t \in \{1,2,3,4\}$ representere de observerte tilstandene for et individ i. En overgang fra tilstand r til tilstand s for $r \neq s$, skjer med intensiteten $q_{rs}(t)$, der $q_{rs}(t) > 0$ for $(r,s) \in \{(1,2),(1,3),(1,4),(2,3),(2,4),(3,4)\}$, og $q_{rs}(t) = 0$ for $(r,s) \in \{(2,1),(3,1),(4,1),(3,2),(4,2),(4,3)\}$. Lar intensiteten q_{rs} representere det samme som tidligere (se kap. 2.2.1), og jeg får følgende overgangsintensitetsmatrise:

$$\mathbf{Q}(t) = \begin{bmatrix} -(q_{12}(t)+q_{13}(t)+q_{14}(t)) & q_{12}(t) & q_{13}(t) & q_{14}(t) \\ 0 & -(q_{23}(t)+q_{24}(t)) & q_{23}(t) & q_{24}(t) \\ 0 & 0 & -q_{34}(t) & q_{34}(t) \\ 0 & 0 & 0 & 0 \end{bmatrix}.$$

Ved å la tidsavhengigheten bli håndrert ved å bruke en stykkevis konstant tilnærming, betyr det at intensitetene ikke varierer i tidsintervallet $(t,u]$, men at $q_{rs}(t)$ vil være konstant i dette intervallet og bestemt ved tid t.

3.2 Markovmatrisen $\mathbf{P}(t,u)$

Siden jeg har at intensitetene er konstante i et tidsintervall $(t,u]$, kan jeg sette opp overgangssannsynlighetene til fire-tilstandsmodellen ved å bruke Kolmogorovs baklengs differensiallikning. Baklengslikningene er utledet ved å dele opp tidsintervallet $(t,u]$, inn i to perioder $(t,t+dt]$ og $(t+dt,u]$, der dt er positiv og liten, og videre se på overgangene i hver periode separat. Baklengslikningene er et resultat av en første stegsanalyse, der det første steget vil være over det korte tidsintervallet av varighet dt [17], kap. 6. Denne metoden utfører en analyse av de mulige overgangene på første trinn, etterfulgt av en anvendelse av Markovegenskapen.

3.2.1 Overgangssannsynlighetene fra tilstand 3

Antar at en person er observert i tilstand 3 ved tid t. Denne personen kan da ved neste observasjon enten bli observert i den samme, altså tilstand 3, eller i tilstand 4 ved en gitt framtid u. Jeg starter først med å finne sannsynligheten for at personen fremdeles vil være i tilstand 3 ved tid u. For å kunne gjøre det må en betinge på det som skjer i det første, korte tidsintervallet $(t,t+dt]$,

$$\begin{aligned} &\Pr(\text{Tilstand 3 ved tid } u|\text{Tilstand 3 ved tid } t) \\ =&\Pr(\text{Tilstand 3 i } (t,t+dt]) * \Pr(\text{Tilstand 3 i (t+dt,u])} \\ =&(1- \text{ sannsynlighet for å forlate tilstand 3}) * \Pr(\text{Tilstand 3 i } (t+dt,u]). \end{aligned}$$

Dette er det samme som å skrive

$$\begin{aligned} p_{33}(t,u) &= p_{33}(t,t+dt)\ p_{33}(t+dt,u) \\ &= (1-q_{34}dt)\ p_{33}(t+dt,u) + o(dt), \end{aligned}$$

der q_{34} betraktes som en konstant siden den blir bestemt ved tiden t og holdes fast gjennom hele intervallet. Flytter det ene p_{33}-leddet fra høyresiden over på venstresiden, og dividerer deretter med dt på begge sider

$$\frac{p_{33}(t,u)-p_{33}(t+dt,u)}{dt} = -q_{34}\ p_{33}(t+dt,u) + \frac{o(dt)}{dt}.$$

La $dt \to 0$, som medfører at $\frac{o(dt)}{dt} \to 0$. Venstresiden av likningen blir den deriverte av p_{33} med hensyn på t

$$-p'_{33}(t,u) = -q_{34}\ p_{33}(t,u),$$

der minustegnet kommer av at det er en baklengslikning. Ved å dele begge sider med $p_{33}(t,u)$, får jeg en differensiallikning som kan løses

$$-\frac{p'_{33}(t,u)}{p_{33}(t,u)} = -q_{34}.$$

Venstresiden kan skrives om til

$$-\frac{d}{ds}[\ln p_{33}(s,u)] = -q_{34}$$

Integrerer uttrykket fra t til u

$$\begin{aligned} -\int_t^u \frac{d}{ds}[\ln p_{33}(s,u)]ds &= -q_{34}\int_t^u ds \\ -[\ln p_{33}(s,u)]_{s=t}^{s=u} &= -q_{34}(u-t) \end{aligned}$$

Med initialbetingelsen, $p_{33}(u,u) = 1$, får jeg at

$$\ln p_{33}(t,u) = -q_{34}(u-t),$$

siden $\ln p_{33}(u,u) = 0$. Sannsynligheten for å være i tilstand 3, gitt at prosessen startet i tilstand 3, blir da:

$$p_{33}(t,u) = e^{-q_{34}(u-t)}.$$

Ser at med $t < u$ vil sannsynligheten for å være i tilstand 3 avta eksponensielt med økende u, desto større tidsintervallet blir.

Det følger da videre at sannsynligheten for å gå til tilstand 4, gitt at prosessen startet i tilstand 3, blir:

$$p_{34}(t,u) = 1 - p_{33}(t,u) = 1 - e^{-q_{34}(u-t)}.$$

3.2.2 Overgangssannsynlighetene fra tilstand 2

Antar nå at en person er i tilstand 2 ved tid t. Denne personen kan enten være i denne tilstanden ved tid u, eller hoppe til tilstand 3 eller tilstand 4. For å finne overgangssannsynligheten for at personen vil være i tilstand 2 ved en gitt framtid u, kan jeg bruke samme framgangsmåte som ble brukt for å finne $p_{33}(t,u)$. Den eneste forskjellen her er at istedenfor en intensitet har jeg nå to intensiteter, siden det nå er mulig å bevege seg til både tilstand 3 og 4.

$$\begin{aligned} p_{22}(t,u) &= p_{22}(t,t+dt)\ p_{22}(t+dt,u) \\ &= (1-(q_{23}+q_{24})dt)\ p_{22}(t+dt,u) + o(dt) \\ \frac{p_{22}(t,u) - p_{22}(t+dt,u)}{dt} &= -(q_{23}+q_{24})\ p_{22}(t+dt,u) + \frac{o(dt)}{dt}. \end{aligned}$$

Lar $dt \to 0$, som gir differensiallikningen

$$-\frac{p'_{22}(t,u)}{p_{22}(t,u)} = -(q_{23}+q_{24}).$$

Tar integralet fra t til u, og bruker initialbetingelsen $p_{22}(u,u) = 1$, som gir overgangssannsynligheten:

$$p_{22}(t,u) = e^{-(q_{23}+q_{24})(u-t)}.$$

Fra tilstand 2 til tilstand 3

Neste steg er å sette opp uttrykket for overgangssannsynligheten for å hoppe fra tilstand 2 til tilstand 3. Jeg har her to mulige overganger; man kan enten bevege seg fra tilstand 2 til 3 i intervallet $(t, t+dt]$, eller bevege seg fra tilstand 2 til 3 i intervallet $(t+dt, u]$. Jeg kan da sette opp følgende uttrykk for de to mulighetene:

$$\begin{aligned} p_{23}(t,u) &= p_{22}(t,t+dt)\ p_{23}(t+dt,u) + p_{23}(t,t+dt)\ p_{33}(t+dt,u) \\ &= (1-(q_{23}+q_{24})dt)\ p_{23}(t+dt,u) + q_{23}dt\ p_{33}(t+dt,u) + o(dt) \end{aligned}$$

Flytter $p_{23}(t+dt,u)$-leddet fra høyresiden over på venstresiden, og dividerer deretter med dt på begge sider

$$\frac{p_{23}(t,u) - p_{23}(t+dt,u)}{dt} = -(q_{23}+q_{24})\ p_{23}(t+dt,u) + q_{23}\ p_{33}(t+dt,u) + \frac{o(dt)}{dt}$$

Lar $dt \to 0$ og får et uttrykk for den deriverte av p_{23} med hensyn på t

$$-p'_{23}(t,u) = -(q_{23}+q_{24})\ p_{23}(t,u) + q_{23}\ p_{33}(t,u).$$

Skriver litt om på uttrykket, slik at det blir en lineær 1.ordens-differensiallikning

$$-\left(p'_{23}(t,u) - (q_{23}+q_{24})\ p_{23}(t,u)\right) = q_{23}\ p_{33}(t,u).$$

Multipliserer begge sider med den integrerende faktoren, som her er:

$$e^{\int_t^u (q_{23}+q_{24})ds} = e^{(q_{23}+q_{24})(u-t)},$$

og får at uttrykket kan skrives som

$$-\frac{d}{dt}\left(e^{(q_{23}+q_{24})(u-t)}\ p_{23}(t,u)\right) = q_{23}\ p_{33}(t,u)\ e^{(q_{23}+q_{24})(u-t)}.$$

Tar integralet fra t til u på begge sider, og bruker initialbetingelsen $p_{23}(u,u) = 0$

$$e^{(q_{23}+q_{24})(u-t)}\ p_{23}(t,u) = q_{23}\int_t^u p_{33}(\tau,u)\ e^{(q_{23}+q_{24})(u-\tau)}d\tau,$$

flytter eksponensialuttrykket over på høyre side

$$p_{23}(t,u) = q_{23}\ e^{-(q_{23}+q_{24})(u-t)}\int_t^u p_{33}(\tau,u)\ e^{(q_{23}+q_{24})(u-\tau)}d\tau.$$

Setter inn for det uttrykket som jeg fant tidligere for p_{33}, og får at overgangssannsynligheten fra tilstand 2 til 3 blir enten

$$p_{23}(t,u) = q_{23}\ e^{-(q_{23}+q_{24})(u-t)} \int_t^u e^{(q_{23}+q_{24}-q_{34})(u-\tau)} d\tau, \tag{3.1}$$

for $(q_{23} + q_{24} \neq q_{34})$, eller

$$p_{23}(t,u) = (u-t)\ q_{23}\ e^{-(q_{23}+q_{24})(u-t)}. \tag{3.2}$$

for $(q_{23} + q_{24} = q_{34})$.

Ser nå bare på integralet i (3.1) og skriver den ut

$$\begin{aligned}
\int_t^u e^{(q_{23}+q_{24}-q_{34})(u-\tau)} d\tau &= e^{(q_{23}+q_{24}-q_{34})u} \int_t^u e^{-(q_{23}+q_{24}-q_{34})\tau} d\tau \\
&= \frac{-e^{-(q_{23}+q_{24}-q_{34})u}}{q_{23}+q_{24}-q_{34}} \left[e^{-(q_{23}+q_{24}-q_{34})\tau}\right]_t^u \\
&= \frac{-e^{-(q_{23}+q_{24}-q_{34})u}}{q_{23}+q_{24}-q_{34}} \left(e^{-(q_{23}+q_{24}-q_{34})u} - e^{-(q_{23}+q_{24}-q_{34})t}\right) \\
&= \frac{\left(-1 + e^{(q_{23}+q_{24}-q_{34})(u-t)}\right)}{q_{23}+q_{24}-q_{34}}.
\end{aligned}$$

Setter dette inn i (3.1) og får

$$p_{23}(t,u) = \frac{q_{23} e^{-(q_{23}+q_{24})(u-t)} \left(-1 + e^{-(q_{23}+q_{24}-q_{34})(u-t)}\right)}{q_{23}+q_{24}-q_{34}}. \tag{3.3}$$

Ser her at hvis $t = 0$ blir likningene (3.2) og (3.3) lik de overgangssannsynlighetene som ble funnet av Jackson[8], bare med andre intensiteter. Ved å studere dette uttrykket nærmere ser jeg at det alltid vil bli positivt uansett om $q_{23} + q_{24}$ er større eller mindre enn q_{34}.

Fra tilstand 2 til tilstand 4

Videre følger det at sannsynligheten for å gå til tilstand 4, gitt at man starter i tilstand 2, vil bli:

$$p_{24}(t,u) = 1 - p_{22}(t,u) - p_{23}(t,u).$$

3.2.3 Overgangssannsynlighetene fra tilstand 1

Antar nå at en person er i tilstand 1 ved tid t, personen kan her enten hoppe til tilstand 2, 3 eller 4. Starter med å finne sannsynligheten for at personen blir værende i tilstand 1 ved tid u. Jeg kan igjen bruke samme framgangsmåte som ble brukt for å finne $p_{33}(t,u)$ og $p_{22}(t,u)$, men her med tre intensiteter siden det er tre mulige overganger.

$$\begin{aligned}
p_{11}(t,u) &= p_{11}(t,t+dt)\ p_{11}(t+dt,u) \\
&= (1 - (q_{12}+q_{13}+q_{14})dt)\ p_{11}(t+dt,u) + o(dt) \\
\frac{p_{11}(t,u) - p_{11}(t+dt,u)}{dt} &= -(q_{12}+q_{13}+q_{14})\ p_{11}(t+dt,u) + \frac{o(dt)}{dt}.
\end{aligned}$$

Lar $dt \to 0$, som gir likningen

$$-\frac{p'_{11}(t,u)}{p_{11}(t,u)} = -(q_{12}+q_{13}+q_{14}).$$

Tar integralet fra t til u, og bruker initialbetingelsen $p_{11}(u,u) = 1$, som gir overgangssannsynligheten:

$$p_{11}(t,u) = e^{-(q_{12}+q_{13}+q_{14})(u-t)}.$$

Fra tilstand 1 til tilstand 2

Overgangssannsynligheten for å hoppe fra tilstand 1 til tilstand 2 kan settes opp ved å bruke samme framgangsmåte som ble brukt for å finne $p_{23}(t,u)$.

$$\begin{aligned} p_{12}(t,u) &= p_{11}(t,t+dt)\ p_{12}(t+dt,u) + p_{12}(t,t+dt)\ p_{22}(t+dt,u) \\ &= (1-(q_{12}+q_{13}+q_{14})dt)\ \ p_{12}(t+dt,u) + q_{12}dt\ p_{22}(t+dt,u) + o(dt), \end{aligned}$$

skriver om på uttrykket og får

$$\frac{p_{12}(t,u) - p_{12}(t+dt,u)}{dt} = -(q_{12}+q_{13}+q_{14})\ p_{12}(t+dt,u) + q_{12}\ p_{22}(t+dt,u) + \frac{o(dt)}{dt}.$$

Lar $dt \to 0$, som gir likningen

$$-p'_{12}(t,u) = -(q_{12}+q_{13}+q_{14})\ p_{12}(t,u) + q_{12}\ p_{22}(t,u),$$

ved å flytte det ene leddet over til venstresiden får jeg en differensiallikning som kan løses

$$-\left(p'_{12}(t,u) - (q_{12}+q_{13}+q_{14})\ p_{12}(t,u)\right) = q_{12}\ p_{22}(t,u).$$

Multipliserer begge sider med den integrerende faktor

$$e^{\int_t^u (q_{12}+q_{13}+q_{14})ds} = e^{(q_{12}+q_{13}+q_{14})(u-t)},$$

slik at uttrykket blir da

$$-[\frac{d}{dt}(e^{(q_{12}+q_{13}+q_{14})(u-t)}\ p_{12}(t,u))] = q_{12}\ p_{22}(t,u)\ e^{(q_{12}+q_{13}+q_{14})(u-t)}.$$

Integrerer fra t til u på begge sidene, og bruker initialbetingelsen $p_{12}(u,u) = 0$, og får

$$p_{12}(t,u) = q_{12}\ e^{-(q_{12}+q_{13}+q_{14})(u-t)} \int_t^u p_{22}(\tau,u)\ e^{(q_{12}+q_{13}+q_{14})(u-\tau)} d\tau.$$

Setter inn for p_{22} i uttrykket, og får at overgangssannsynligheten enten blir

$$p_{12}(t,u) = q_{12}\ e^{-(q_{12}+q_{13}+q_{14})(u-t)} \int_t^u e^{(q_{12}+q_{13}+q_{14}-q_{23}-q_{24})(u-\tau)} d\tau. \tag{3.4}$$

for $(q_{12}+q_{13}+q_{14} \neq q_{23}+q_{24})$, eller

$$p_{12}(t,u) = (u-t)q_{12}\ e^{-(q_{12}+q_{13}+q_{14})(u-t)}. \tag{3.5}$$

for $(q_{12}+q_{13}+q_{14}=q_{23}+q_{24})$.

Legger merke til at integralet i (3.4) er på samme form som i likning (3.1), slik at likning (3.4) kan bli skrevet som

$$p_{12}(t,u) = q_{12}\ e^{-(q_{12}+q_{13}+q_{14})(u-t)}\frac{\left(-1+e^{(q_{12}+q_{13}+q_{14}-q_{23}-q_{24})(u-t)}\right)}{q_{12}+q_{13}+q_{14}-q_{23}-q_{24}}. \tag{3.6}$$

Ved å sammenligne (3.5) og (3.6) med henholdsvis (3.2) og (3.3), ser jeg at de er på samme form, men med hver sine tilhørende intensiteter.

Fra tilstand 1 til tilstand 3

Overgangssannsynligheten for å bevege seg fra tilstand 1 til tilstand 3 kan her også regnes ut med samme framgangsmåte som ble brukt for $p_{23}(t,u)$ og $p_{12}(t,u)$, men det er for denne situasjonen mulig å bevege seg innom tilstand 2 før en ender opp i tilstand 3. Uttrykket som jeg får her vil da se slik ut, med et ekstra ledd:

$$\begin{aligned} p_{13}(t,u) &= p_{11}(t,t+dt)\ p_{13}(t+dt,u) + p_{13}(t,t+dt)\ p_{33}(t+dt,u) \\ &\quad + p_{12}(t,t+dt)\ p_{23}(t+dt,u) \\ &= (1-(q_{12}+q_{13}+q_{14})\ dt)\ \ p_{13}(t+dt,u) + \ q_{13}dt\ p_{33}(t+dt,u) \\ &\quad + q_{12}dt\ p_{23}(t+dt,u) + o(dt). \end{aligned}$$

Flytter det ene leddet over på venstresiden og dividerer med dt

$$\begin{aligned} \frac{p_{13}(t,u)-p_{13}(t+dt,u)}{dt} &= -(q_{12}+q_{13}+q_{14})\ p_{13}(t+dt,u) + q_{13}\ p_{33}(t+dt,u) \\ &\quad + q_{12}\ p_{23}(t+dt,u) + \frac{o(dt)}{dt}. \end{aligned}$$

Lar $dt \to 0$, som gir

$$-p'_{13}(t,u) = -(q_{12}+q_{13}+q_{14})\ p_{13}(t,u) + q_{13}\ p_{33}(t,u) + q_{12}\ p_{23}(t,u),$$

og skriver uttrykket om til en differensiallikning

$$-\left(p'_{13}(t,u) - (q_{12}+q_{13}+q_{14})\ p_{13}(t,u)\right) = q_{13}\ p_{33}(t,u) + q_{12}\ p_{23}(t,u).$$

Multipliserer begge sider med den integrerende faktor

$$e^{\int_t^u (q_{12}+q_{13}+q_{14})ds} = e^{(q_{12}+q_{13}+q_{14})(u-t)}$$

og får uttrykket

$$\begin{aligned} -\left[\frac{d}{dt}\left(e^{(q_{12}+q_{13}+q_{14})(u-t)}\ p_{13}(t,u)\right)\right] &= q_{13}\ p_{33}(t,u)e^{(q_{12}+q_{13}+q_{14})(u-t)} \\ &\quad + q_{12}\ p_{23}(t,u)e^{(q_{12}+q_{13}+q_{14})(u-t)}. \end{aligned}$$

Tar integralet fra t til u på begge sidene, og bruker initialbetingelsen $p_{13}(u,u) = 0$, og får

$$\begin{aligned} p_{13}(t,u) &= q_{13}e^{-(q_{12}+q_{13}+q_{14})(u-t)} \int_t^u p_{33}(\tau,u)e^{(q_{12}+q_{13}+q_{14})(u-\tau)}d\tau \\ &\quad +q_{12}e^{-(q_{12}+q_{13}+q_{14})(u-t)} \int_t^u p_{23}(\tau,u)e^{(q_{12}+q_{13}+q_{14})(u-\tau)}d\tau. \end{aligned}$$

Setter inn for p_{33} og p_{23} i uttrykket, og får

$$\begin{aligned} p_{13}(t,u) &= q_{13}\ e^{-(q_{12}+q_{13}+q_{14})(u-t)} \int_t^u e^{(q_{12}+q_{13}+q_{14}-q_{34})(u-\tau)}d\tau \\ &\quad + q_{12}\ e^{-(q_{12}+q_{13}+q_{14})(u-t)} \int_t^u \frac{q_{23}}{q_{23}+q_{24}-q_{34}} \\ &\quad * e^{-(q_{23}+q_{24})(u-\tau)} \left(-1 + e^{-(q_{23}+q_{24}-q_{34})(u-\tau)}\right) e^{(q_{12}+q_{13}+q_{14})(u-\tau)}d\tau. \end{aligned}$$

Uttrykket som jeg har fått er en del større og mer komplisert enn tidligere. Det første leddet kjenner jeg igjen fra de tidligere uttrykkene og representerer det direkte hoppet fra tilstand 1 til tilstand 3. Det andre leddet er derimot nytt i forhold til før, og dette leddet kommer av at prossessen har vært innom tilstand 2 på vei til tilstand 3. Jeg har nå fått et mer komplisert uttrykk inn i integralet som ikke er like hensiktmessig å utlede.

Fra tilstand 1 til 4

Til slutt følger det at sannsynligheten for å gå til tilstand 4, gitt at man er i tilstand 1, blir:

$$p_{14}(t,u) = 1 - p_{11}(t,u) - p_{12}(t,u) - p_{13}(t,u).$$

Kapittel 4

Likelihooden til fire-tilstandsmodellen

4.1 Enkeltpersonenes bidrag til likelihooden

Likelihooden kan kalkuleres ut fra Markovmatrisen $\mathbf{P}(t,u)$, som for dette eksempelet er

$$\mathbf{P} = \begin{bmatrix} p_{11}(t,u) & p_{12}(t,u) & p_{13}(t,u) & p_{14}(t,u) \\ 0 & p_{22}(t,u) & p_{23}(t,u) & p_{24}(t,u) \\ 0 & 0 & p_{33}(t,u) & p_{34}(t,u) \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Antar at en person har observasjoner $X_{t_1}, \ldots, X_{t_m}$ ved tidene $t_1, \ldots, t_m$. Betrakter to påfølgende observasjoner, X_{t_j} og $X_{t_{j+1}}$, ved tidene t_j og t_{j+1}. Bidraget til likelihooden fra disse to tilstandene er:

$$L_j = \Pr(X_{t_{j+1}} | X_{t_j}) = p_{X_{t_j}, X_{t_{j+1}}}(t_j, t_{j+1}), \tag{4.1}$$

der likelihooden avhenger av overgangsintensitetsmatrisen $\mathbf{Q}$, som ble brukt for å utlede $\mathbf{P}(t,u)$. Fra tidligere har jeg at intensitetene i $\mathbf{Q}$ kan være tidsavhengige via kovariater, men de vil ikke ha noen innvirkning på det generelle uttrykket for likelihooden. De kommer først inn når man skal sette opp de eksplisitte uttrykkene for likelihooden. Bidraget til likelihooden vil (ifølge Jackson[8]) bli litt annerledes for dødstilstanden. Det er vanlig at dødstidspunktet er kjent, mens tilstanden like før er ukjent. Legger merke til at tolkningen av tid t_m vil bli litt annerledes i tilfellet ved død. Tid t_m representerer i denne situasjonen den eksakte overgangstiden og ikke den siste observeringstiden. Hvis $X_{t_m} = 4$, blir bidraget til likelihooden summert over den ukjente tilstanden k like før døden:

$$L_m = \sum_{k \neq 4} p_{X_{t_{m-1}}, k}(t_{m-1}, t_m)\ q_{k,4}, \tag{4.2}$$

der summen tas over alle mulige tilstander k som kan bli besøkt mellom $X_{t_{m-1}}$ og tilstand 4. Likelihooden blir også litt annerledes i tilfellet med sensurerte tilstander der tilstandens eksakte verdi er ukjent, men kjent for å være i et bestemt intervall [8]. F.eks. i overlevelsesanalyse er et dødstidspunkt høyre-sensurert hvis studien avsluttes og pasienten fremdeles er i live, siden dødstidspunktet er kjent for å være større

enn avslutningstiden. I multitilstandsmodeller for midlertidlige observerte prosesser, er tiden for endring av tilstander vanligvis intervallsensurert kjent til å være innenfor avgrensede intervaller. Dette fører til en likelihood basert på likning (4.1) [8]. Tilstander kan i noen tilfeller også være sensurerte så vel som hendelsestidspunkter. F.eks. ved slutten av noen kroniske sykdomsstudier, er pasienter kjent for å være i live, men i en ukjent tilstand. For en slik sensurert observasjon $S(t_m)$, bare kjent til å være i en tilstand i en mengde C, er det ekvivalente bidraget til likelihooden

$$L_m = \sum_{k \in C} p_{X_{t_{m-1}},k}(t_{m-1}, t_m). \tag{4.3}$$

Denne spesielle likelihooden er ikke nødvendig hvis tilstanden er kjent ved slutten av studiet. Selv om overlevelsestiden er sensurert, er tilstanden ved slutten av studiet ikke sensurert.

Den fulle likelihooden L for denne personen er produktet av alle slike ledd L_j over alle overgangene

$$L = \prod_{j=1}^{m} L_j = \Pr(X_{t_1}, \ldots, X_{t_m}).$$

Bruker Markovegenskapen og får

$$L = \Pr(X_{t_1}) \Pr(X_{t_2}|X_{t_1}) \times \cdots \times \Pr(X_{t_m}|X_{t_{m-1}}),$$

der $\Pr(X_{t_{j+1}}|X_{t_j}), j = 1, \ldots, m-1$, er overgangssannsynlighetene.

4.2 Eksempeluttrykk i likelihooden

Jeg har nå et generelt uttrykk for likelihooden og kan nå sette opp forskjellige uttrykk for likelihooden. Starter med å se på et enkelt tilfelle der jeg antar at en person er observert med overgangene $1 \to 2 \to 3$ ved tidene t_1, t_2, t_3. Bidraget til likelihooden fra $X_{t_1} = 1$ og $X_{t_2} = 2$ blir

$$L_1 = \Pr(X_{t_2}|X_{t_1}) = p_{X_{t_1},X_{t_2}}(t_1, t_2) = p_{12}(t_1, t_2)$$

Siden jeg har en observasjon ved tiden t_m trenger jeg ikke å behandle denne tilstanden som sensurert og jeg får at bidraget til likelihooden fra $X_{t_2} = 2$ og $X_{t_3} = 3$ blir

$$L_2 = \Pr(X_{t_3}|X_{t_2}) = p_{X_{t_2},X_{t_3}}(t_2, t_3) = p_{23}(t_2, t_3)$$

Den fulle likelihooden blir da

$$\begin{aligned} L &= \Pr(X_{t_1}) \Pr(X_{t_2}|X_{t_1}) \Pr(X_{t_3}|X_{t_2}) \\ &= p_{12}(t_1, t_2) * p_{23}(t_2, t_3), \end{aligned} \tag{4.4}$$

der $\Pr(X_{t_1})$ vil være lik 1 siden alle individene i dette studiet skal begynne i tilstand 1.

Setter inn for de uttrykkene som allerede er funnet for overgangssannsynlighetene

(se kap. 3). For å skille intensitetene som inngår i disse uttrykkene velger jeg å bruke notasjonene q_1 og q_2. For $(q_{1.12}+q_{1.13}+q_{1.14} \neq q_{1.23}+q_{1.24})$ og $(q_{2.23}+q_{2.24} \neq q_{2.34})$ blir likelihooden

$$\begin{aligned} L =& \frac{q_{1.12}\ e^{-(q_{1.12}+q_{1.13}+q_{1.14})(t_2-t_1)}\left(-1+e^{(q_{1.12}+q_{1.13}+q_{1.14}-q_{1.23}-q_{1.24})(t_2-t_1)}\right)}{q_{1.12}+q_{1.13}+q_{1.14}-q_{1.23}-q_{1.24}} \\ & * \frac{q_{2.23}e^{-(q_{2.23}+q_{2.24})(t_3-t_2)}\left(-1+e^{-(q_{2.23}+q_{2.24}-q_{2.34})(t_3-t_2)}\right)}{q_{2.23}+q_{2.24}-q_{2.34}}, \end{aligned}$$

og for $(q_{1.12}+q_{1.13}+q_{1.14} = q_{1.23}+q_{1.24})$ og $(q_{2.23}+q_{2.24} = q_{2.34})$ blir likelihooden

$$\begin{aligned} L =& (t_2-t_1)q_{1.12}\ e^{-(q_{1.12}+q_{1.13}+q_{1.14})(t_2-t_1)} \\ & * (t_3-t_2)\ q_{2.23}\ e^{-(q_{2.23}+q_{2.24})(t_3-t_2)}. \end{aligned}$$

Det er viktig å huske at intensitetene kan være tidsavhengige via kovariater, og at de da bestemmes i begynnelsen av hvert tidsintervall. Dette medfører at de intensitetene som inngår i p_{12} er bestemt ved tid t_1, mens de intensitetene som inngår i p_{23} vil være bestemt ved tid t_2. Dersom intensitetene avhenger bare av en ikke-tidsavhengig kovariat, som kjønn, vil intensitetene være de samme, bestemt i tid t_1, og man kan da sette $q_1 = q_2 = q$.

Videre vil jeg sette opp likelihooden der den siste observasjonen er dødstilstanden. Antar nå at en person er observert med overgangene $1 \to 2 \to 4$ ved tidene t_1, t_2, t_3. Bidraget til likelihooden fra $X_{t_2} = 2$ og $X_{t_3} = 4$ blir

$$\begin{aligned} L_2 &= \sum_{k\neq 4} p_{X_{t_2},k}(t_2,t_3)\ q_{k,4} = \sum_{k\neq 4} p_{2,k}(t_2,t_3)\ q_{k,4} \\ &= p_{22}(t_2,t_3)\ q_{24} + p_{23}(t_2,t_3)\ q_{34}. \end{aligned}$$

Den fulle likelihooden blir da

$$\begin{aligned} L &= \Pr(X_{t_1})\Pr(X_{t_2}|X_{t_1})\Pr(X_{t_3}|X_{t_2}) \\ &= p_{12}(t_1,t_2) * \left(p_{22}(t_2,t_3)\ q_{24} + p_{23}(t_2,t_3)\ q_{34}\right). \end{aligned}$$

For $(q_{1.12}+q_{1.13}+q_{1.14} \neq q_{1.23}+q_{1.24})$ og $(q_{2.23}+q_{2.24} \neq q_{2.34})$ vil likelihooden være:

$$\begin{aligned} L =& \frac{q_{1.12}\ e^{-(q_{1.12}+q_{1.13}+q_{1.14})(t_2-t_1)}\left(-1+e^{(q_{1.12}+q_{1.13}+q_{1.14}-q_{1.23}-q_{1.24})(t_2-t_1)}\right)}{q_{1.12}+q_{1.13}+q_{1.14}-q_{1.23}-q_{1.24}} \\ & * \Bigg(q_{2.24}\ e^{-(q_{2.23}+q_{2.24})(t_3-t_2)} \\ & + \frac{q_{2.34}\ q_{2.23}e^{-(q_{2.23}+q_{2.24})(t_3-t_2)}\left(-1+e^{-(q_{2.23}+q_{2.24}-q_{2.34})(t_3-t_2)}\right)}{q_{2.23}+q_{2.24}-q_{2.34}}\Bigg), \end{aligned}$$

og for $(q_{1.12} + q_{1.13} + q_{1.14} = q_{1.23} + q_{1.24})$ og $(q_{2.23} + q_{2.24} = q_{2.34})$ blir likelihooden

$$\begin{aligned} L =& (t_2 - t_1) q_{1.12}\ e^{-(q_{1.12}+q_{1.13}+q_{1.14})(t_2-t_1)} \\ & * \left(q_{2.24}\ e^{-(q_{2.23}+q_{2.24})(t_3-t_2)} + (t_3 - t_2)\ q_{2.34}\ q_{2.23}\ e^{-(q_{2.23}+q_{2.24})(t_3-t_2)} \right). \end{aligned}$$

4.2.1 Oppsummering

De uttrykkene som jeg har fått for likelihooden er både store og kompliserte, og jeg har her bare tatt med tre observasjoner. Dersom jeg hadde tatt med flere observasjoner ville den bare blitt enda større og mer uoversiktig. Jeg ser derfor ut fra disse likelihoodene at det vil bli vanskelig å regne noe mer analytisk på uttrykkene, og at den bør bli løst numerisk. Jackson m. fl. [10] har utviklet en pakke til den statistiske programvaren `R` som de har kalt *msm*. Denne pakken inneholder en del funksjoner, blant annet `msm()` som er i stand til å modellere multitilstandsmodeller, der den estimerer maksimum likelihood estimatene. Jeg vil i kapittel 6 ta for meg hva msm-pakken er og litt om hvordan den fungerer.

Kapittel 5

Feilklassifisering

Jeg vil i dette kapittelet utvide multitilstandsmodellen ved å inkludere feilklassifisering. Lar de observerte datasettene være bestemt av en sannsynlighetsfordeling som er betinget på de uobserverte tilstandene, og den underliggende Markovmodellen bestemmes ut fra en overgangsintensitetsmatrise $\mathbf{Q}$ som tidligere.

5.1 Utvidelse av spesialtilfellet

Jeg velger å utvide modellen ved å tillate feilklassifisering av tre tilstander. Lar det nå være mulig å feilklassifisere tilstand 1 til å være i tilstand 2, tilstand 2 til å være i enten tilstand 1 eller 3, og tilstand 3 til å være i tilstand 2. For $t \geq 0$ lar jeg $X_t \in \{1, 2, 3, 4\}$ være de sanne tilstandene for et individ, og lar $X_t^* \in \{1, 2, 3, 4\}$ være de observerte tilstandene. Med de mulige feilklassifiseringssannsynlighetene $c_{rs} = \Pr(X_t^* = s | X_t = r)$ gir det matrisen

$$\mathbf{C} = \begin{bmatrix} c_{11} & c_{12} & c_{13} & c_{14} \\ c_{21} & c_{22} & c_{23} & c_{24} \\ c_{31} & c_{32} & c_{33} & c_{34} \\ c_{41} & c_{42} & c_{43} & c_{44} \end{bmatrix} = \begin{bmatrix} 1 - c_{12} & c_{12} & 0 & 0 \\ c_{21} & 1 - c_{21} - c_{23} & c_{23} & 0 \\ 0 & c_{32} & 1 - c_{32} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Siden tilstand 4, døden, er målt uten feilklassifisering, og rekkene summeres til 1, er det bare fire parameter som skal estimeres: c_{12}, c_{21}, c_{32} og c_{23}. Lar overgangsintensitetsmatrisen $\mathbf{Q}(t)$ være den samme som tidligere.

5.2 Maksimum likelihood estimering

Antar at en person har observasjoner ved tidene $t_1, \ldots, t_m$. Jeg får her at bidraget av denne personen til likelihooden, når feilklassifisering blir inkludert, blir likning (2.1) som var

$$\begin{aligned} L = &\sum_{X_{t_1}} \Pr(X_{t_1}^* | X_{t_1}) \Pr(X_{t_1}) \sum_{X_{t_2}} \Pr(X_{t_2}^* | X_{t_2}) \Pr(X_{t_2} | X_{t_1}) \\ &\cdots \sum_{X_{t_m}} \Pr(X_{t_m}^* | X_{t_m}) \Pr(X_{t_m} | X_{t_{m-1}}), \end{aligned} \tag{5.1}$$

der $\Pr(X_t^*|X_t)$ er feilklassifiseringssannsynlighetene, og $\Pr(X_{t_{j+1}}|X_{t_j}), j = 1, \ldots, m-1$ er de overgangssannsynlighetene som ble utledet tidligere (se kap.3). I tilfellet med feilklassifisering vil ikke $\Pr(X_{t_1} = i)$ lenger være lik 1 for $i = 1, 2, 3, 4$. Siden det er X_t^* som er den observerte og ikke X_t, kan det nå være tilfeller der den observerte tilstanden er tilstand 1, mens den sanne tilstanden faktisk er tilstand 2. For å finne ut hvordan kan estimere fordelingen av den første tilstanden, $\Pr(X_{t_1})$, trenger jeg å se litt nærmere på logistisk regresjon.

5.2.1 Logistisk regresjon

Dobson and Barnett [5] har satt opp den generelle logistiske regresjonsmodellen for to kategorier:

$$\text{logit}(\pi_i) = \log\left(\frac{\pi_i}{1-\pi_i}\right) = \mathbf{z}_i^T\beta,$$

der π_i representerer suksessannsynligheten, $\mathbf{z}_i$ er en vektor av kontinuerlige målinger som tilsvarer kovariater og dummy-variabler som tilsvarer faktornivåer, og β er parametervektoren. Denne modellen blir svært mye brukt for å analysere data med binære eller binomiske responser og flere forklaringsvariabler.

Logistisk regresjon kan utvides til å kunne håndtere kategoriske responser som er *polytom*, altså responser med mer enn to kategorier. Et alternativ baserer seg på generaliseringer av logistisk regresjon fra *dikotome* responser, med to kategorier, til nominale (bestående av ikke-ordnede kategorier) og ordinale (bestående av ordnede kategorier) responser med mer enn to kategorier. For nominal eller ordinal logistisk regresjon vil en av de målte eller observerte kategoriske variablene bli betraktet som responsen, mens alle de andre variablene vil bli betraktet som forklaringsvariabler.

Setter først opp den multinominale fordelingen som legger grunnlaget for å modellere kategoriske data med mer enn to kategorier. Betrakter en tilfeldig variabel Y med J kategorier. Lar $\pi_1, \pi_2, \ldots, \pi_J$ være de respektive sannsynlighetene, med $\pi_1 + \pi_2 + \cdots + \pi_J = 1$. Hvis det er n uavhengige observasjoner av Y som resulterer i y_1 utfall i kategori 1, y_2 utfall i kategori 2, osv, la da

$$\mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_J \end{bmatrix}, \qquad \text{med} \qquad \sum_{j=1}^{J} y_j = n.$$

Den multinominale fordelingen er

$$f(\mathbf{y}|n) = \frac{n!}{y_1!y_2!\ldots y_J!}\pi_1^{y_1}\pi_2^{y_2}\ldots\pi_J^{y_J},$$

med $\text{E}(Y_j) = n\pi_j$, $\text{var}(Y_j) = n\pi_j(1-\pi_j)$.

Nominal logistisk regresjon er brukt når der er ingen naturlig orden blant responskategoriene. En kategori blir vilkårlig valgt til å være referansekategorien, og det er

vanlig å anta at dette er den første kategorien. Logiten for de andre kategoriene er da definert ved

$$\text{logit}(\pi_j) = \log\left(\frac{\pi_j}{\pi_1}\right) = \mathbf{z}_j^T\beta_j, \qquad \text{for } j = 2,\ldots,J. \tag{5.2}$$

For å estimere parameterene β_j, blir de $(J-1)$ logit likningene brukt simultant. Når man har fått tak i parameterestimatene $\mathbf{b}_j$, kan de lineære prediktorene $\mathbf{z}_j^T\mathbf{b}_j$ regnes ut. Likning (5.2) gir

$$\hat{\pi}_j = \hat{\pi}_1 \exp(\mathbf{z}_j^T\mathbf{b}_j), \qquad \text{for } j = 2,\ldots,J,$$

og $\hat{\pi}_1 + \hat{\pi}_2 + \cdots + \hat{\pi}_J = 1$, slik at

$$\hat{\pi}_1 = \frac{1}{1+\sum_{j=2}^{J}\exp(\mathbf{z}_j^T\mathbf{b}_j)}$$

og

$$\hat{\pi}_j = \frac{\exp(\mathbf{z}_j^T\mathbf{b}_j)}{1+\sum_{j=2}^{J}\exp(\mathbf{z}_j^T\mathbf{b}_j)}, \qquad \text{for } j = 2,\ldots,J.$$

5.2.2 Fordelingen av den første tilstanden i likelihooden

I tilfellet med fire tilstander kan modellen bestå av tre responsekategorier, henholdsvis tilstand 1, 2 og 3. Det er da lettest å anvende den nominale regresjonsmodellen siden den ikke krever flere forutsetninger, mens den ordinale krever en del flere forutsetninger og er mye mer innviklet å anvende. Jeg får at

$$\Pr(X_{t_1}=1) = \frac{1}{1+\sum_{j=2}^{3}\exp(\gamma_j^T\mathbf{z}_j(t_1))} = \frac{1}{1+\exp(\gamma_2^T\mathbf{z}_2(t_1))+\exp(\gamma_3^T\mathbf{z}_3(t_1))}$$

$$\Pr(X_{t_1}=2) = \frac{\exp(\gamma_2^T\mathbf{z}_2(t_1))}{1+\exp(\gamma_2^T\mathbf{z}_2(t_1))+\exp(\gamma_3^T\mathbf{z}_3(t_1))}$$

$$\Pr(X_{t_1}=3) = \frac{\exp(\gamma_3^T\mathbf{z}_3(t_1))}{1+\exp(\gamma_2^T\mathbf{z}_2(t_1))+\exp(\gamma_3^T\mathbf{z}_3(t_1))}$$

I tillegg har jeg at $\Pr(X_{t_1}=4)=0$.

5.2.3 Matrisebidrag til likelihooden

Matriser kan bli brukt til å beskrive likelihooden på en lettere måte enn med bruk av summasjoner. På matriseform blir det individuelles bidrag til likelihooden

$$L = \mathbf{f}\mathbf{T}_2\mathbf{T}_3 \times \cdots \times \mathbf{T}_m\mathbf{1}, \tag{5.3}$$

der $\mathbf{1} = (1,1,1,1)^T$, $\mathbf{f}$ er en 1×4 rekkevektor, $\mathbf{T}_j$ er en 4×4 matrise for $j = 2,\ldots,m$. Uttrykket vil generelt se ut som følgende med vektorer og matriser

$$\Rightarrow L = \begin{bmatrix} - & - & - & - \end{bmatrix} \begin{bmatrix} - & - & - & - \\ - & - & - & - \\ - & - & - & - \\ - & - & - & - \end{bmatrix} \times \cdots \times \begin{bmatrix} - & - & - & - \\ - & - & - & - \\ - & - & - & - \\ - & - & - & - \end{bmatrix} \begin{bmatrix} - \\ - \\ - \\ - \end{bmatrix}.$$

Multiplikasjon av en rekkevektor og en matrise kan bli definert som følgende, når $\mathbf{x}$ er en $1 \times n$ og A er $n \times m$:

$$\mathbf{x}A = \begin{bmatrix} x_1 & x_2 & \dots & x_n \end{bmatrix} \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1m} \\ a_{21} & a_{22} & \dots & a_{2m} \\ \vdots & & & \\ a_{n1} & a_{n2} & \dots & a_{nm} \end{bmatrix}$$

$$= \begin{bmatrix} (x_1a_{11} + x_2a_{21} + \cdots + x_ma_{m1}) & \dots & (x_1a_{1r} + \cdots + x_ma_{mr}) \end{bmatrix}.$$

En vektor kan bare multipliseres med en matrise dersom antall kolonner i $\mathbf{x}$ er lik antall rekker i A. Produktet av en rekkevektor og en matrise er en ny rekkevektor. I dette tilfellet vil en multiplikasjon mellom en 1×3 vektor og en 3×3 matrise gi en ny 1×3 vektor, som videre kan brukes til å multipliseres med neste matrise. Matrisebidraget vil etter multiplikasjonen med $\mathbf{T}_m$ stå igjen med en rekkevektor og $\mathbf{1}$. Produktet av en 1×3 rekkevektor og en 3×1 vektor er en 1×1 matrise, som blir kalt for skalarproduktet av vektorene eller indre produktet.

5.2.4 Matrisebidraget for denne likelihooden

Lar $\mathbf{f}$ være en 1×4 rekkevektor med det $r'te$ elementet $\Pr(X^*_{t_1}|X_{t_1} = r)\Pr(X_{t_1} = r)$, for $r \in \{1, 2, 3, 4\}$, som gir den generelle vektoren

$$\mathbf{f} = \begin{bmatrix} \Pr(X^*_{t_1}|X_{t_1} = 1)\Pr(X_{t_1} = 1) \\ \Pr(X^*_{t_1}|X_{t_1} = 2)\Pr(X_{t_1} = 2) \\ \Pr(X^*_{t_1}|X_{t_1} = 3)\Pr(X_{t_1} = 3) \\ \Pr(X^*_{t_1}|X_{t_1} = 4)\Pr(X_{t_1} = 4) \end{bmatrix}^T.$$

Jeg har i dette eksempelet bestemt at for den første observasjonen kan den observerte tilstanden bare være tilstand 1, $X^*_{t_1} = 1$, og at den sanne tilstanden bare kan være tilstand 1 eller tilstand 2. Med dette har jeg da bare to responskategorier og velger dermed å anvende den logistiske regresjonsmodellen akkurat slik som van den Hout m. fl. [19]. Jeg får da

$$\mathbf{f} = \begin{bmatrix} \Pr(X^*_{t_1}|X_{t_1} = 1)\Pr(X_{t_1} = 1) \\ \Pr(X^*_{t_1}|X_{t_1} = 2)(1 - \Pr(X_{t_1} = 1)) \\ 0 \\ 0 \end{bmatrix}^T.$$

For tidene $t_2, \dots, t_{m-1}$, la $\mathbf{T}_j$ være en 4×4 matrise med (r, s)-elementet

$$\Pr(X^*_{t_j}|X_{t_j} = s)\Pr(X_{t_j} = s|X_{t_{j-1}} = r),$$

som for dette eksempelet blir

$$\mathbf{T}_j = \begin{bmatrix} \Pr(X^*_{t_j}|X_{t_j} = 1)p_{11}(t_{j-1}, t_j) & \dots & \Pr(X^*_{t_j}|X_{t_j} = 4)p_{14}(t_{j-1}, t_j) \\ \Pr(X^*_{t_j}|X_{t_j} = 1)p_{21}(t_{j-1}, t_j) & \dots & \Pr(X^*_{t_j}|X_{t_j} = 4)p_{24}(t_{j-1}, t_j) \\ \Pr(X^*_{t_j}|X_{t_j} = 1)p_{31}(t_{j-1}, t_j) & \dots & \Pr(X^*_{t_j}|X_{t_j} = 4)p_{34}(t_{j-1}, t_j) \\ \Pr(X^*_{t_j}|X_{t_j} = 1)p_{41}(t_{j-1}, t_j) & \dots & \Pr(X^*_{t_j}|X_{t_j} = 4)p_{44}(t_{j-1}, t_j) \end{bmatrix},$$

der

$$\Pr(X^*_{t_j}|X_{t_j}=1) = (1-c_{12})\mathbf{1}(X^*_{t_j}=1) + c_{12}\mathbf{1}(X^*_{t_j}=2)$$
$$\Pr(X^*_{t_j}|X_{t_j}=2) = c_{21}\mathbf{1}(X^*_{t_j}=1) + (1-c_{21}-c_{23})\mathbf{1}(X^*_{t_j}=2) + c_{23}\mathbf{1}(X^*_{t_j}=3)$$
$$\Pr(X^*_{t_j}|X_{t_j}=3) = c_{32}\mathbf{1}(X^*_{t_j}=2) + (1-c_{32})\mathbf{1}(X^*_{t_j}=3)$$
$$\Pr(X^*_{t_j}|X_{t_j}=4) = \mathbf{1}(X^*_{t_j}=4)$$

I tilfellet med død, $X^*_{t_m}=4$, er (r,s)-elementet av $\mathbf{T}_m$ lik $\Pr(X_{t_m}=s|X_{t_{m-1}}=r)q_{s4}(t_m)$. Det er antatt at et individ er i en ukjent latent tilstand s ved tid t_m, og så en umiddelbar død,

$$\mathbf{T}_m = \begin{bmatrix} p_{11}(t_{m-1},t_m)\ q_{14}(t_m) & \dots & p_{14}(t_{m-1},t_m)\ q_{44}(t_m) \\ p_{21}(t_{m-1},t_m)\ q_{14}(t_m) & \dots & p_{24}(t_{m-1},t_m)\ q_{44}(t_m) \\ p_{31}(t_{m-1},t_m)\ q_{14}(t_m) & \dots & p_{34}(t_{m-1},t_m)\ q_{44}(t_m) \\ p_{41}(t_{m-1},t_m)\ q_{14}(t_m) & \dots & p_{44}(t_{m-1},t_m)\ q_{44}(t_m) \end{bmatrix}$$

$$= \begin{bmatrix} p_{11}(t_{m-1},t_m)\ q_{14}(t_m) & p_{12}(t_{m-1},t_m)\ q_{24}(t_m) & p_{13}(t_{m-1},t_m)\ q_{34}(t_m) & 0 \\ 0 & p_{22}(t_{m-1},t_m)\ q_{24}(t_m) & p_{23}(t_{m-1},t_m)\ q_{34}(t_m) & 0 \\ 0 & 0 & p_{33}(t_{m-1},t_m)\ q_{34}(t_m) & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}.$$

I tilfellet med sensurering, er det kjent at personen fremdeles er i live, men at tilstanden er ukjent. I dette tilfellet er (r,s)-elementet av $\mathbf{T}_m$

$$\Pr(X_{t_m}=s|X_{t_{m-1}}=r)\mathbf{1}_{[s\in\{1,2,3\}]},$$

der $1_{[A]}$ er lik 1 dersom A er sann og 0 ellers. T_m vil da se slik ut:

$$\mathbf{T}_m = \begin{bmatrix} p_{11}(t_{m-1},t_m) & p_{12}(t_{m-1},t_m) & p_{13}(t_{m-1},t_m) & 0 \\ 0 & p_{22}(t_{m-1},t_m) & p_{23}(t_{m-1},t_m) & 0 \\ 0 & 0 & p_{33}(t_{m-1},t_m) & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}.$$

5.2.5 Eksempel

Antar nå at en person er observert med overgangene $1 \to 2 \to 4$ ved tidene t_1, t_2, t_3. Bidraget til likelihooden blir

$$L = \mathbf{f}\mathbf{T}_2\mathbf{T}_3,$$

der

$$\mathbf{f} = \begin{bmatrix} \Pr(X^*_{t_1}=1|X_{t_1}=1)\Pr(X_{t_1}=1) \\ \Pr(X^*_{t_1}=1|X_{t_1}=2)\Pr(X_{t_1}=2) \\ \Pr(X^*_{t_1}=1|X_{t_1}=3)\Pr(X_{t_1}=3) \\ \Pr(X^*_{t_1}=1|X_{t_1}=4)\Pr(X_{t_1}=4) \end{bmatrix}^T = \begin{bmatrix} c_{11}\Pr(X_{t_1}=1) \\ c_{21}\Pr(X_{t_1}=2) \\ 0 \\ 0 \end{bmatrix}^T.$$

$$\mathbf{T}_2 = \begin{bmatrix} \Pr(X^*_{t_2}=2|X_{t_2}=1)p_{11}(t_1,t_2) & \dots & \Pr(X^*_{t_2}=2|X_{t_2}=4)p_{14}(t_1,t_2) \\ \Pr(X^*_{t_2}=2|X_{t_2}=1)p_{21}(t_1,t_2) & \dots & \Pr(X^*_{t_2}=2|X_{t_2}=4)p_{24}(t_1,t_2) \\ \Pr(X^*_{t_2}=2|X_{t_2}=1)p_{31}(t_1,t_2) & \dots & \Pr(X^*_{t_2}=2|X_{t_2}=4)p_{34}(t_1,t_2) \\ \Pr(X^*_{t_2}=2|X_{t_2}=1)p_{41}(t_1,t_2) & \dots & \Pr(X^*_{t_2}=2|X_{t_2}=4)p_{44}(t_1,t_2) \end{bmatrix}$$

$$= \begin{bmatrix} c_{12}\ p_{11}(t_1,t_2) & c_{22}\ p_{22}(t_1,t_2) & c_{32}\ p_{13}(t_1,t_2) & 0 \\ 0 & c_{22}\ p_{22}(t_1,t_2) & c_{32}\ p_{23}(t_1,t_2) & 0 \\ 0 & 0 & c_{32}\ p_{33}(t_1,t_2) & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix},$$

$$\mathbf{T}_3 = \begin{bmatrix} p_{11}(t_2,t_3)\ q_{14}(t_3) & \dots & p_{14}(t_2,t_3)\ q_{44}(t_3) \\ p_{21}(t_2,t_3)\ q_{14}(t_3) & \dots & p_{24}(t_2,t_3)\ q_{44}(t_3) \\ p_{31}(t_2,t_3)\ q_{14}(t_3) & \dots & p_{34}(t_2,t_3)\ q_{44}(t_3) \\ p_{41}(t_2,t_3)\ q_{14}(t_3) & \dots & p_{44}(t_2,t_3)\ q_{44}(t_3) \end{bmatrix}$$

$$= \begin{bmatrix} p_{11}(t_2,t_3)\ q_{14}(t_3) & p_{12}(t_2,t_3)\ q_{24}(t_m 3) & p_{13}(t_2,t_3)\ q_{34}(t_3) & 0 \\ 0 & p_{22}(t_2,t_3)\ q_{24}(t_3) & p_{23}(t_2,t_3)\ q_{34}(t_3) & 0 \\ 0 & 0 & p_{33}(t_2,t_3)\ q_{34}(t_3) & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}.$$

5.3 Oppsummering

Jeg ser med én gang ut fra eksempelet at ved å tillate feilklassifisering av noen av tilstandene får jeg mye mer å forholde meg til. For å estimere maksimum likelihood estimatene kreves det her at det må brukes en numerisk metode. Pakken `msm` som er utviklet av Jackson m. fl. [10] er også i stand til å modellere skjulte Markovmodeller, som jeg vil ta for meg i neste kapittel.

Kapittel 6

Aktuell programvare i R

6.1 Beskrivelse av msm

For å beregne likelihooden i denne oppgaven har jeg brukt den statistiske programvaren `R`. I 2003 ble det av Jackson m. fl. [10] utviklet en pakke kalt *msm* for `R` som tillater en generell multitilstandsmodell å bli tilpasset til data med observasjoner over tid. Med de funksjonene som finnes i msm gir det meg mulighetene for å modellere overgangsratene og skjulte Markovmodeller i forhold til kovariater, og muligheten for å modellere datasett med ulike observasjonsordninger, inkludert tilstander som er sensurerte. Manualen som er skrevet av Jackson [8] gir en god innføring i teorien som ligger bak multitilstands Markov og skjulte Markovmodeller, og viser til typisk bruk av msm-pakken.

6.1.1 msm-pakken

De uttrykkene som jeg har utarbeidet for likelihooden i de tidligere kapitlene, med eller uten feilklassifisering, er både lange og kompliserte. For å kunne estimere overgangsintensitetene må en maksimere likelihooden numerisk. msm er en pakke med funksjoner for modellering av multitilstandsmodeller, og selve funksjonen `msm()` implementerer maksimum likelihood estimering for generelle multitilstands Markov eller skjulte Markovmodeller i kontinuerlig tid [8]. Denne funksjonen skal være i stand til å estimere en hvilken som helst form av overgangsintensitetsmatriser og feilklassifiseringsmatriser, med ubegrenset antall kovariater på hver av disse matrisene [10].

`msm()` tilpasser en tidskontinuerlig Markov eller skjult Markov multitilstandsmodell med maksimum likelihood. Observasjoner av prosessen kan skje ved vilkårlige tider, eller så kan de eksakte overgangstidene mellom tilstandene være kjente. Det kan også bli tilpasset kovariater til overgangsintensiteter tilhørende Markovkjeden eller til den skjulte Markovobservasjonsprosessen [7]. Med passende argumenter vil `msm()` returnere en liste med viktige resultater av modelltilpassingen, inkludert parameterestimatene og deres kovarianser. For å vise maksimum likelihood estimatene og 95% konfidensintervallene, er det bare å skrive navnet på objektet i kommandovinduet i `R` [8].

Msm er utformet for å kunne tilpasses tidskontinuerlige Markovmodeller, prosesser der overganger mellom tilstander kan oppstå når som helst. Disse modellene er definert ut fra intensiteter, som både dekker tiden brukt i den nåværende tilstanden og sannsynlighetene for det neste steget. For enkle tidskontinuerlige multitilstands Markovmodeller, beregnes likelihooden i forhold til overgangsintensitetsmatrisen $\mathbf{Q}$. Dersom de eksakte overgangstidene ikke er kjent beregnes likelihooden ved å bruke overgangsmatrisen $\mathbf{P}(t) = \exp(t\mathbf{Q})$, der exp er den *eksponensielle matrisen* [7]. For skjulte Markovmodeller, beregnes likelihooden for et individ med k observasjoner, direkte ved å summere over den ukjente tilstanden for hver observasjon, slik at det produseres et produkt med k matriser.

Et krav til datasettet er at det må være nok informasjon på hver tilstand for å kunne estimere hver av overgangsratene, ellers vil likelihooden bli flat og maksimum blir ikke funnet. Det kan også være viktig for optimaliseringen å velge et passende sett av initialverdier [7].

6.2 Argumenter som inngår i msm()

`msm()` kan ta inn mange forskjellige argumenter ut fra hvordan modellen blir definert. Multitilstandsmodellen som skal tilpasses datasettet defineres i `qmatrix`-argumentet. Modellen styres av overgangsintensitetsmatrisen $\mathbf{Q}$, så for å fortelle `msm` hvilke overganger som er tillatt må det defineres en matrise som er av samme form og størrelse som $\mathbf{Q}$. Den skal ha 0 i de (r, s)-elementene der det ikke er tillatt å hoppe fra tilstand r til tilstand s, og en initialverdi der en overgang er lov. Elementene i diagonalen blir ignorert, siden den er begrenset til å være lik minus summen av resten av rekken [7].

For å kunne modellere feilklassifisering av tilstander, må det defineres en feilklassifiseringsmatrise $\mathbf{C}$ i `ematrix`-argumentet. Rekkene av denne matrisen korresponderer til sanne tilstander og kolonnene til observerte tilstander. For å fortelle `msm` hvilke tilstander som kan klassifiseres feil må det defineres en matrise av samme størrelse som $\mathbf{C}$. De elementene i matrisen der feilklassifisering ikke er tillatt skal være 0, og diagonalelementene vil også her bli ignorert i `ematrix`, siden rekkene er begrenset til å være lik 1 [8].

Argumentet `obstype` er en vektor som spesifiserer observasjonsordningen for hver observasjon i datasettet. Den kan enten inkluderes i datasettet som en egen kolonne, eller bli satt lik et enkelt tall i kallet på `msm` som da vil bety at alle observasjonene i datasettet er antatt å være av denne typen. De elementene som kan være i `obstype` er enten 1, 2 eller 3, som betyr følgende: Tallet 1 indikerer en observasjon av prosessen ved en vilkårlig tid, og at tilstandene mellom observasjonstidene er ukjente. Tallet 2 indikerer en eksakt overgangstid med tilstanden fra forrige observasjon beholdt opptil den nåværende observasjonen. Tallet 3 indikerer også en eksakt overgangstid, men tilstanden i det øyeblikket en går inn i denne tilstanden er ukjent. Et eksempel på dette er dødstilstanden der dødstidspunktet ofte er en eksakt tid, mens tilstanden før døden inntreffer kan være ukjent. Ved å spesifisere observasjons-

typene i en vektor trenger ikke `obstype` bare å variere mellom individer, men kan også variere mellom tilstander etter hverandre for samme individ. Hvis `obstype` ikke er spesifisert er denne defaultet til å være 1 for alle observasjonene [7]. Argumentet `obstrue` kan bli spesifisert dersom det er feilklassifiseringer i datasettet. Den skal være en vektor av logiske eller numeriske verdier som forteller hvilke observasjoner (`TRUE,1`) som er observasjoner av den underliggende tilstanden uten feil, og hvilke (`FALSE,0`) som er realiseringer av en skjult Markovmodell.

`msm` gir muligheten for å spesifisere observasjonstypene på flere måter ved å ha andre argumenter som omhandler nesten det samme. Argumentet `exacttimes` antar ved default at overgangene i Markovprosessen skjer ved ukjente anledninger mellom observasjonstidene. Hvis `exacttimes=TRUE` betyr det at alle observasjonstidene antas å representere de eksakte og komplette tidene av overgangene i prosessen, som vil være ekvivalent til at hver av rekkene i datasettet har `obstype=2`. Er både `obstype` og `exacttimes` spesifisert, vil `exacttimes` bli ignorert. Argumentet `death` angir en absorberende tilstand der overgangstiden til tilstanden er kjent nøyaktig, og individet antas å være i en transient tilstand, altså i live, like før. Dersom tilstanden er kjent mellom dødstidspunktet og forrige observasjon, bør `obstype=2` spesifiseres for dødstiden, eller `exacttimes=TRUE` hvis tilstanden er kjent til enhver tid, som medfører at argumentet `death` blir ignorert. `obstype` er en generalisering av argumentene `death` og `exacttimes` for å tillate ulike ordninger på hver observasjon. Den vil også overstyre både `death` og `exacttimes` [7].

Jackson m. fl. [10] sier at for å maksimere likelihooden i den skjulte Markovmodellen kan det brukes en quasi-Newton optimaliseringsalgoritme, beskrevet av Dennis og Schnabel [4] kap. 9, som ikke krever spesifikasjonene av de deriverte av funksjonen for å bli maksimert. En quasi-Newton algoritme kan spesifiseres med argumentet `method=" BFGS"` som ofte kan være raskere og mer robust enn defaulten, Nelder-Mead [8]. BFGS-metoden er en quasi-Newton metode oppkalt etter Broyden, Fletcher, Goldfarb og Shanno i 1970. Quasi-Newton's metoder er basert på Newton's metode for å finne stasjonærpunktet av en funksjon, der gradienten er 0. Den antar at funksjonen kan bli lokalt approksimert som en kvadrert funksjon i området rundt optimum, og den bruker første og andre deriverte for å finne stasjonærpunktet. Nelder-Mead metoden var utarbeidet av John Nelder og Roger Mead i 1965 [14], og er en metode for å minimere en funksjon i et flerdimensjonelt rom [12], kap. 11. Det er en direkte søkemetode og den fortsetter uten å ta med noen derivater av funksjonen og uten linjesøk. Den er relativt treg, men fungerer bra for ikke-deriverbare funksjoner.

6.3 Enkel testing av msm med datasettet cav

For å få et lite innblikk av hvordan `msm`-funksjonen fungerer, har jeg valgt å gå gjennom de kodene Jackson har kommet med i manualen [8]. Ved å bruke de kommandosetningene han har kommet med får jeg nøyaktig de samme resultatene som er blitt gitt i manualen. Når han har tilpasset feilklassifiseringsmodellen med `msm` har han brukt følgende kode:

```
> oneway4.q <- rbind(c(0,0.148,0,0.0171),c(0,0,0.202,0.081),
+                    c(0,0,0,0.126),c(0,0,0,0))
> ematrix <- rbind(c(0,0.1,0,0),c(0.1,0,0.1,0),
+                  c(0,0.1,0,0),c(0,0,0,0))
> cavmisc.msm <- msm(state ~ years, subject=PTNUM, data=cav,
+                    qmatrix=oneway4.q, ematrix=ematrix, death=4,
+                    obstrue=firstobs, method="BFGS")
```

Jeg begynner med å sjekke om det er noen begrensninger på de initialverdiene som sendes inn i funksjonen, og om hvilke endringer som er tillatt å gjøre på modellene. Starter med å se på hvilke initialverdier som kan sendes inn i `qmatrix`. Tabell 6.1 viser et utvalg av de ulike valgene av `oneway4.q` som er blitt testet på dette datasettet. Ut fra denne tabellen, ser det ut til at hvis verdiene blir valgt til å være mellom 0 og opp til 1, og i tillegg ikke varierer så mye fra hverandre, er sjansen størst for å ikke få advarsel og verdier som er omtrentlig de samme som utgangspunktet. Velges verdiene til å være litt større enn 1, eller ved å la størrelsene på verdiene variere litt fra hverandre, er sjansen stor for at man vil få en advarsel og noen ville estimater, som enten er tilnærmet lik 0 eller veldig stort. Hvis verdiene velges til å være enda litt høyere, medfører det i de fleste tilfellene at programmet vil gi error. Av de kjøringene som ikke fikk advarsel, er det bare én som skiller seg ut, nemlig de andre initialverdiene som ble testet ut. De har gitt litt andre verdier, og den klarer ikke å estimere q_{24}. Resultatene fra denne kjøringen ligner mest på de resultatene fra tilfellene der jeg får en advarsel, ved og i tillegg studere initialverdiene ligner også disse på hverandre. I de kjøringene som ga advarsel får man bare ut estimatene uten de tilhørende konfidensintervallene. Jeg vil i kapittel 7 gå nærmere inn på hva advarselen sier, men en mulig årsak for at det går galt ved estimeringen kan være konvergensproblemer. Den eneste rimelige endringen på modellen ut fra dette datasettet, er eventuelt å inkludere en intensitet til, nemlig q_{13}, som ved kjøring av `msm` bare gir tilsvarende resultat som utgangspunktet.

Jeg vil nå se på hvilke initialverdier som kan sendes inn i `ematrix`. Tabell 6.2 viser resultatene av noen av de initialverdiene som ble testet på dette datasettet. Kan her huske på den ene egenskapen til klassifiseringsmatrisen som er at rekkene skal summeres til 1. De første initialverdiene som testes ut, er for å se hva som skjer dersom jeg velger verdiene slik at estimatene på rekkene i matrisen blir større enn 1. Velges enten c_{12} eller c_{32} til å være større enn 1 medfører det at de vil bli estimert til å være null, som betyr at tilstand 1 eller 3 ikke kan bli feilklassifisert på lik linje med tilstand 4. Velges c_{21} og c_{23} til å være større enn 1, medfører det at jeg får to error og en advarsel. Ser ut fra tabellen at hvis verdiene velges slik at ingen av rekkene blir større enn 1, får jeg i de fleste tilfellene ingen advarsler, og verdier som er omtrentlig de samme som utgangspunktet. De endringene som kan gjøres på feilklassifiseringsmatrisen er å inkludere feilklassifiseringsestimatene c_{13} og/eller c_{31}. Tabell 6.3 viser resultatene for de valgte endringene. Observerer her at `msm` gir litt forskjellige verdier for hvilken modell en velger, noe som kanskje kan være rimelig siden jeg tillatter flere feilklassifiseringsmuligheter.

Det neste som kan testes ut er hva som skjer når jeg endrer på de andre argument-

oneway4.q												
q_{12}, q_{14}, q_{23}, q_{24}, q_{34}	$\hat{q}_{12}$	$\hat{q}_{14}$	$\hat{q}_{23}$	$\hat{q}_{24}$	$\hat{q}_{34}$	$\hat{c}_{12}$	$\hat{c}_{21}$	$\hat{c}_{23}$	$\hat{c}_{32}$		
0.148, 0.0171, 0.202, 0.081, 0.126	0.08963	0.04136	0.25864	0.03331	0.30758	0.02690	0.17491	0.06318	0.11510		
4, 5, 1, 1, 3	0.08768	0.04304	0.29623	0.00000	0.33028	0.02743	0.17702	0.06138	0.13653		
0.1, 0.1, 0.1, 0.1, 0.1	0.08963	0.04137	0.25876	0.03315	0.30777	0.02686	0.17506	0.06310	0.11518		
1, 1, 1, 1, 1	0.08962	0.04136	0.25879	0.03321	0.30766	0.02691	0.17483	0.06318	0.11517		
2, 2, 2, 2, 2	0.12625	0.00000	0.00000	0.26877	8.4e+23	0.03937	0.07805	0.37825	0.00000	*	
1.5, 1.5, 1.5, 1.5, 1.5	0.12625	0.00000	0.26877	0.00000	1.2e+11	0.03937	0.07805	0.37825	0.00038	*	
1, 2, 3, 1, 2	0.13820	0.00000	0.16317	0.24291	0.23556	0.02313	0.22252	0.06713	0.10327	*	
0.01, 0.01, 0.01, 0.01, 0.01	0.08964	0.04137	0.25842	0.03337	0.30767	0.02695	0.17469	0.06317	0.11509		
0.9, 0.1, 0.5, 0.6, 0.3	0.08766	0.04305	0.29636	0.00000	0.33022	0.02750	0.17677	0.06148	0.13649	*	
0.1, 0.9, 0.1, 0.9, 0.9	0.12625	0.00000	0.26877	0.00000	5.9e+07	0.03937	0.07805	0.37825	0.00000	*	
0.9, 0.9, 0.9, 0.9, 0.9	0.08963	0.04136	0.25872	0.03325	0.30765	0.02691	0.17493	0.06316	0.11513		
0.4, 0.5, 0.1, 0.1, 0.3	0.08963	0.04136	0.25870	0.03326	0.30763	0.02691	0.17491	0.06317	0.11510		
0.4, 0.2, 0.5, 0.5, 0.1	0.08963	0.04136	0.25869	0.03326	0.30763	0.02691	0.17491	0.06318	0.11510		
9, 9, 9, 9, 9	-	-	-	-	-	-	-	-	-	**	
5, 5, 5, 5, 5	-	-	-	-	-	-	-	-	-	**	
3, 3, 3, 3, 3	-	-	-	-	-	-	-	-	-	**	
1, 2, 3, 1, 2	0.13820	0.00000	0.16317	0.24291	0.23556	0.02313	0.22252	0.06713	0.10327	*	
1, 3, 3, 2, 1	0.13813	0.00000	0.16338	0.24315	0.23552	0.02319	0.22228	0.06719	0.10352	*	
2, 4, 1, 1, 2	0.12625	0.00000	0.26877	0.00000	7.9e+07	0.03937	0.07805	0.37825	1.00000	*	
1, 2, 3, 4, 5	-	-	-	-	-	-	-	-	-	**	
1, 4, 3, 2, 1	-	-	-	-	-	-	-	-	-	**	
5, 4, 3, 2, 1	-	-	-	-	-	-	-	-	-	**	

Tabell 6.1: *Oversikt over de ulike initialverdiene som ble testet med de tilhørende estimerte intensitetene, der * eller ** betyr at den valgte initialmatrisen henholdsvis enten gir en advarsel eller en error ved kjøring.*

ematrix c_{12}, c_{21}, c_{23}, c_{32}	$\hat{q}_{12}$	$\hat{q}_{14}$	$\hat{q}_{23}$	$\hat{q}_{24}$	$\hat{q}_{34}$	$\hat{c}_{12}$	$\hat{c}_{21}$	$\hat{c}_{23}$	$\hat{c}_{32}$	
0.1, 0.1, 0.1, 0.1	0.08963	0.04136	0.25864	0.03331	0.30758	0.02690	0.17491	0.06318	0.11510	
1.5, 0.2, 0.6, 0.1	0.11494	0.04137	0.19936	0.02877	0.30910	0.00000	0.33803	0.04010	0.14581	
0.6, 0.2 ,0.6, 1	0.08673	0.04040	0.19531	0.06164	0.34181	0.03193	0.12163	0.14081	0.00000	
0.1, 0.8, 0.4, 0.1	-	-	-	-	-	-	-	-	-	***
0.1, 0.1, 0.2, 0.2	0.08962	0.04136	0.25871	0.03325	0.30763	0.02691	0.17490	0.06318	0.11511	
0.2, 0.2, 0.5, 0.2	0.08963	0.04134	0.25833	0.03366	0.30709	0.02689	0.17481	0.06323	0.11485	
0.6, 0.2, 0.5, 0.7	0.08962	0.04136	0.25871	0.03326	0.30762	0.02691	0.17487	0.06317	0.11510	
0.7, 0.2, 0.6, 0.6	0.11312	0.04332	0.23117	0.00001	0.33315	0.00000	0.34589	0.03845	0.17806	*
0.6, 0.2 ,0.6, 0.6	0.08962	0.04135	0.25862	0.03336	0.30756	0.02691	0.17482	0.06321	0.11505	

Tabell 6.2: *Oversikt over de ulike initialverdiene som ble testet med de tilhørende estimerte intensitetene, der * eller *** betyr at den valgte initialmatrisen henholdsvis enten gir en advarsel eller to error og en advarsel ved kjøring.*

ematrix	$\hat{q}_{12}$	$\hat{q}_{14}$	$\hat{q}_{23}$	$\hat{q}_{24}$	$\hat{q}_{34}$	$\hat{c}_{12}$	$\hat{c}_{13}$	$\hat{c}_{21}$	$\hat{c}_{23}$	$\hat{c}_{31}$	$\hat{c}_{32}$
c_{12}, c_{21}, c_{23}, c_{32}:											
	0.08963	0.04136	0.25864	0.03331	0.30758	0.02690		0.17491	0.06318		0.11510
c_{12}, c_{13}, c_{21}, c_{23}, c_{32}:											
	0.08716	0.04145	0.26842	0.03317	0.30693	0.02962	0.00072	0.15435	0.06284		0.11322
c_{12}, c_{21}, c_{23}, c_{31}, c_{32}:											
	0.09040	0.04140	0.28324	0.02943	0.28882	0.02568		0.18014	0.02233	0.01867	0.11658
c_{12}, c_{13}, c_{21}, c_{23}, c_{31}, c_{32}:											
	0.08690	0.04159	0.30291	0.02733	0.28773	0.02941	0.00095	0.15046	0.01723	0.01866	0.11811

Tabell 6.3: *Endringer på feilklassifiseringsmatrisen, der alle initialverdiene settes til å være 0.1.*

	1	2	3	4	5
$\hat{q}_{12}$	0.08963	0.05427	0.06202	0.08659	0.09856
$\hat{q}_{14}$	0.04136	0.05061	0.04911	0.04745	0.04672
$\hat{q}_{23}$	0.25864	0.09512	0.09365	0.23282	0.20132
$\hat{q}_{24}$	0.03331	0.14895	0.14501	0.06329	0.06220
$\hat{q}_{34}$	0.30758	0.20331	0.20434	0.36906	0.36710
$\hat{c}_{12}$	0.02690	0.06258	0.02831	0.02807	0.00808
$\hat{c}_{21}$	0.17491	0.05488	0.07927	0.16660	0.23797
$\hat{c}_{23}$	0.06318	0.27304	0.22754	0.06384	0.05117
$\hat{c}_{32}$	0.11510	0.01925	0.02386	0.10536	0.11281

Tabell 6.4: *Resulatene fra kjøringene, der (1) viser verdiene fra Jackson's kjøring, (2) viser verdiene med obstype=2 og obstrue=firstobs, (3) viser verdiene med bare argumentet obstype=2, (4) viser verdiene med obstype=1 og obstrue=firstobs, og (5) viser verdiene med bare argumentet obstype=1.*

ene i kallet på `msm`. I denne testingen bruker jeg de matrisene, `qmatrix` og `ematrix`, som var valgt i utgangspunktet. Tabell 6.4 viser resultatene fra kjøringene, der jeg har spesifiert ulike observasjonstyper. Først har jeg testet ut argumentet `obstype=2`, med og uten `obstrue`-argumentet. Siden `obstype` overskriver `death`, er den fjernet fra kjøringene. Deretter har jeg sjekket hva jeg får ved å bruke defaulten `obstype=1`, med og uten `obstrue`-argumentet. Verdiene som kommer ut blir litt forskjellig i alle tilfellene, slik at det ser ut til at valg av observasjonstype faktisk har en liten innvirkning på de estimatene som en får fra `msm`. Kan ut fra tabellen se at tolkningen av feilklassifiseringsestimatene, der `obstype=2`, vil bli litt annerledes enn for de andre. Det kan i dette tilfellet se ut som at sannsynligheten for å feilklassifisere tilstand 2 til å være tilstand 3, er en del høyere enn de andre feilklassifiseringsmulighetene. I de andre tilfellene ser det ut til at det er en høyere sannsynlighet for å klassifisere feil en tilstand til å være en tilstand lavere enn en tilstand høyere.

6.4 Oppsummering

Inntrykket jeg får av `msm` fra manualen og den enkle testingen er at det, for dette datasettet, fungerer stort sett fint dersom man velger rimelige initialverdier. Det eneste som en bør være oppmerksom på er valg av observasjonstyper. Jackson har gjort pakken mer brukervennlig med manualen [8], og han har i tillegg laget ganske utfyllende hjelpebeskrivelser inne i programvaren `R`. I de neste kapitlene skal jeg generere egne datasett, for så å sjekke om `msm` fungerer like godt på dem, eller om det da oppstår problemer.

Kapittel 7

Testing av msm

I dette kapittelet skal jeg prøve ut `msm` på genererte datasett. Begynner først med å lage et enkelt datasett og sjekker hva som skal til for at `msm` klarer å estimere rimelige intensiteter. Deretter vil jeg i neste kapittel, der metoden skal testes, generere datasett med ulike kovariater og feilklassifiseringer, for så å sjekke om `msm` klarer å modellere de utvidede modellene.

7.1 Simulering av datasett

For å kunne bruke `msm` trenger jeg et datasett som er spesifisert som en serie av observasjoner, gruppert etter individer. Jackson [8] påpeker at det, som et minstekrav, må være en dataramme bestående av variabler som gir informasjon om observasjonstidene, den observerte tilstanden av prosessen og identifikasjonsnummerene på deltakerne. Dersom identifikasjonsnummerene mangler, vil alle observasjonene antas å være fra samme individ. Jeg skal her generere et datasett som skal bestå av identifikasjonsnummer (ptnum), alder, tiden de har vært med i studien, den observerte tilstanden, kjønn, og to aldersgrupper. Aldersgruppene skal være to dummy variabler kalt gruppe2 og gruppe3, der gruppe2 er lik 1 dersom individet er 72-73 år gammel, mens gruppe3 er lik 1 dersom individet er 74-76 år gammel. Hvis begge er lik 0 er individet 70-71 år gammel.

For å produsere et datasett har jeg først laget en funksjon: `xMatrise()`, som kan ta inn antall individer og en overgangsintensitetsmatrise Q. Funksjonen har også muligheten til å ta inn flere intensitetsmatriser, slik at jeg i neste kapittel kan generere datasett ut fra flere ulike overgangsintensitetsmatriser, se vedlegg A. Antar i dette tilfellet at alle individene som er med i datasettet, ved inngangen av studien, er 70 år og frisk (tilstand 1), og lar kjønn være tilfeldig trekt. Videre trekkes tiden et individ skal være i den første tilstanden tilfeldig ut fra $\exp(q_{12}+q_{13}+q_{14})$. Individet vil da hoppe tilfeldig til enten tilstand 2, 3 eller 4, med henholdsvis sannsynlighetene: Pr(hoppe til tilstand 2) $= q_{12}/(q_{12}+q_{13}+q_{14})$, Pr(hoppe til tilstand 3) eller Pr(hoppe til tilstand 4). Prosessen fortsetter slik, bare med nye intensiteter, helt til individet er i den absorberende tilstanden døden (tilstand 4). For hvert tidspunkt blir det lagret en rekke i datasettet med informasjon om individet. Funksjonen gjentar dette for alle deltakerne som skal være med i studien. Til slutt vil `xMatrise`

returnere en matrise bestående av ptnum, alder, tid, tilstand, kjønn, gruppe2 og gruppe3, der tid representerer de eksakte overgangstidene mellom tilstandene.

For å transformere datasettet over på en mer ønsket form, har jeg laget en funksjon: `zMatrise()`, som kan ta inn antall individer og den matrisen som ble laget i `xMatrise`-funksjonen. Det er i denne funksjonen at rammene for studien blir bestemt. Velger at studien bare skal gå over 6-7 år, der deltakerne har observasjoner omtrent et år etter forrige observasjon. Dette blir gjort ved å velge et tilfeldig tall mellom 0,9 og 1,1, som her vil tilsvare mellom 10 til 14 måneder, og på denne måten vil observasjonstidene være tilfeldige mellom og innen individene. For hver observasjon fra et individ, vil funksjonen bruke den genererte matrisen for å bestemme hvilken tilstand individet befinner seg i. Har individet hoppet flere ganger mellom observasjonstidene er det den siste tilstanden som vil bli observert. Dersom døden inntreffer før studien er blitt avsluttet, vil den siste observasjonen bli døden (tilstand 4), med den siste observasjonstiden lik dødstidspunktet.

7.2 Størrelse på datasettet

Før jeg skal begynne å utforske funksjonen med ulike argumenter, vil jeg først starte med å teste hvilken betydning omfanget av datasettet har på de estimatene som blir produsert gjennom `msm`-funksjonen. Jeg velger å teste for datasett bestående av 500, 200 og 20 individer, og måle estimatene i de tre tilfellene opp mot hverandre. Jeg gjennomfører 100 simuleringer på hvert av tilfellene, der jeg for hver simulering henter ut og lagrer estimatene i vektorer. Bruker deretter disse estimatene til å regne ut gjennomsnittsverdiene og *mean squared error* (MSE) [3], kap. 7. Lager også histogram av estimatene for å se hvordan de er estimert og sjekke om de ser noen lunde normalfordelte ut. For å lage datasettene har jeg valgt å bruke intensitetsmatrisen:

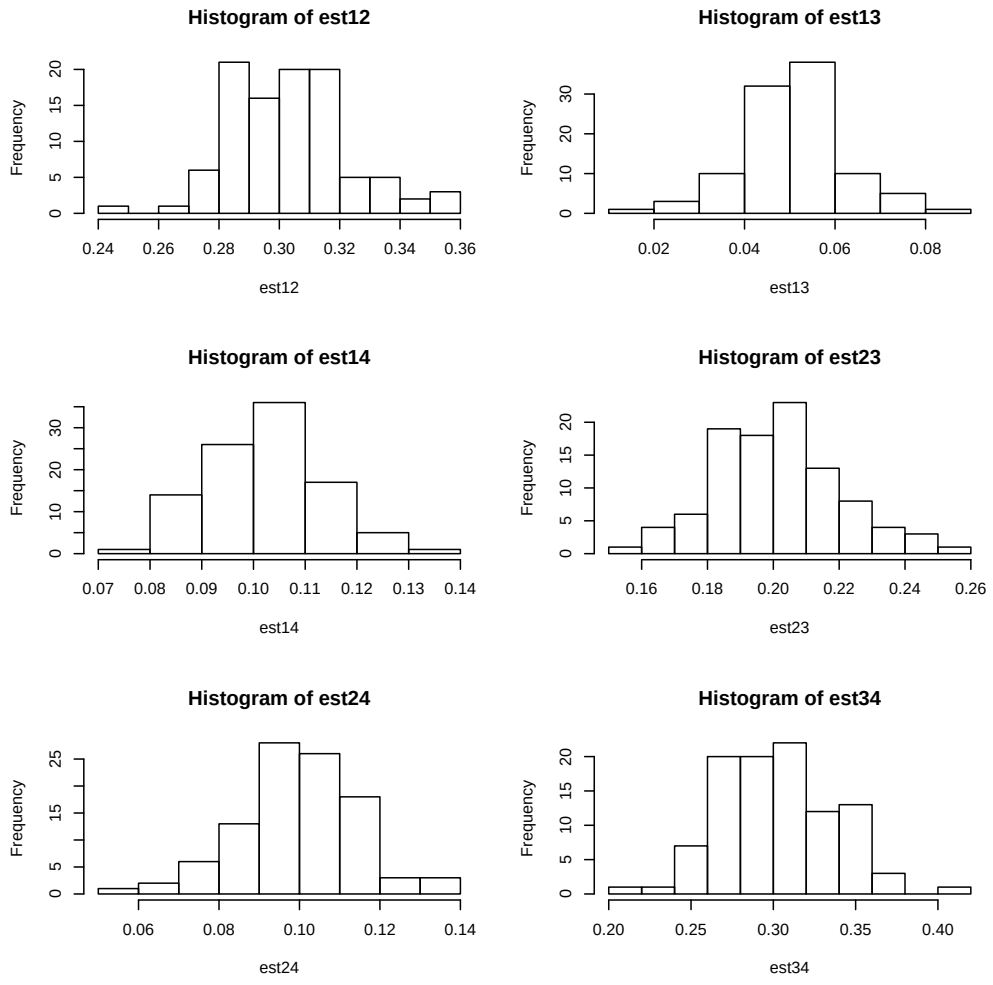
$$\mathbf{Q} = \begin{bmatrix} 0 & 0.3 & 0.05 & 0.1 \\ 0 & 0 & 0.2 & 0.1 \\ 0 & 0 & 0 & 0.3 \\ 0 & 0 & 0 & 0 \end{bmatrix}.$$

Jeg vil ved beregningene av MSE-verdiene bruke denne for å sjekke de estimerte parameterne mot de gitte intensitetene. Deretter kjører jeg så mange som 10 000 simuleringer for hvert av tilfellene slik at man kan få et generelt inntrykk av hvordan estimatene oppfører seg. Regner også her ut gjennomsnittsverdiene, MSE-ene og lager histogrammer av estimatene.

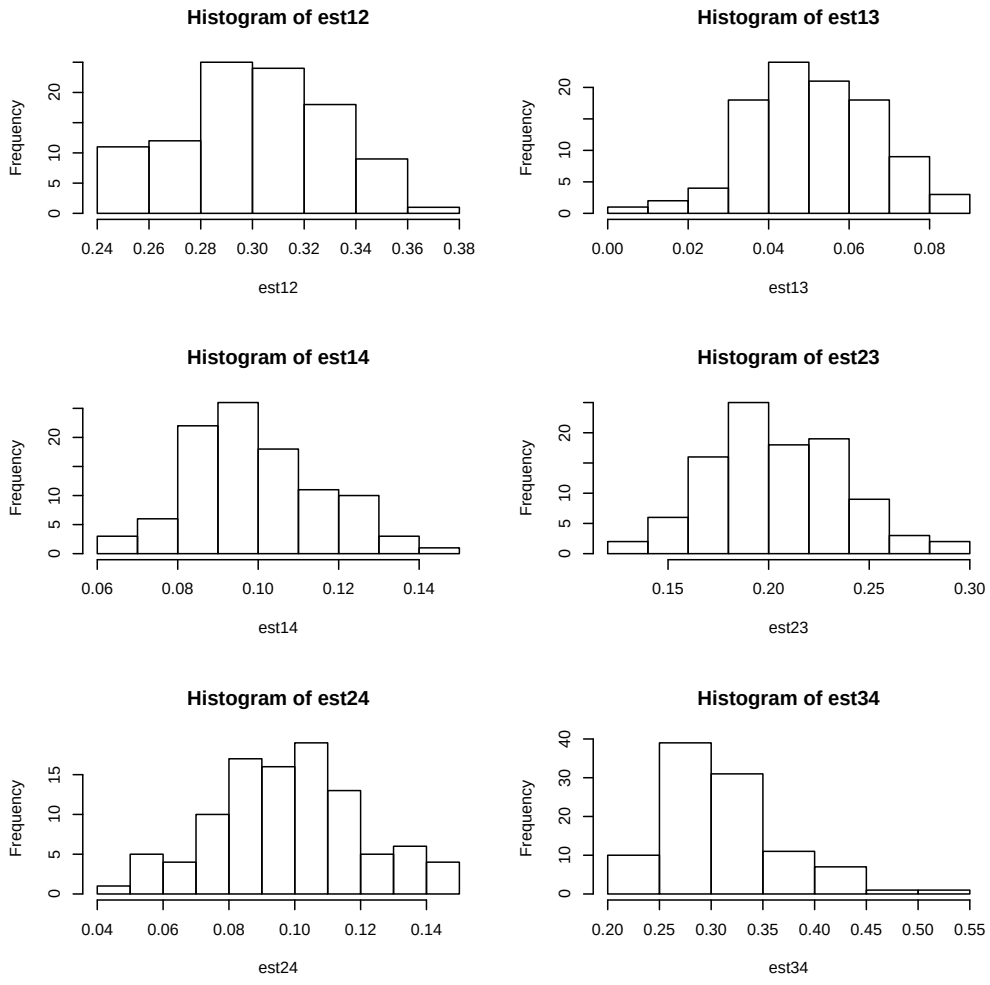
7.2.1 Resultat

Tabellene 7.1 og 7.2 viser resultatene fra de tre kjøringene med henholdsvis gjennomsnittsverdiene og MSE-verdiene, mens figurene 7.1-7.3 viser histogrammene av alle de estimerte estimatene.

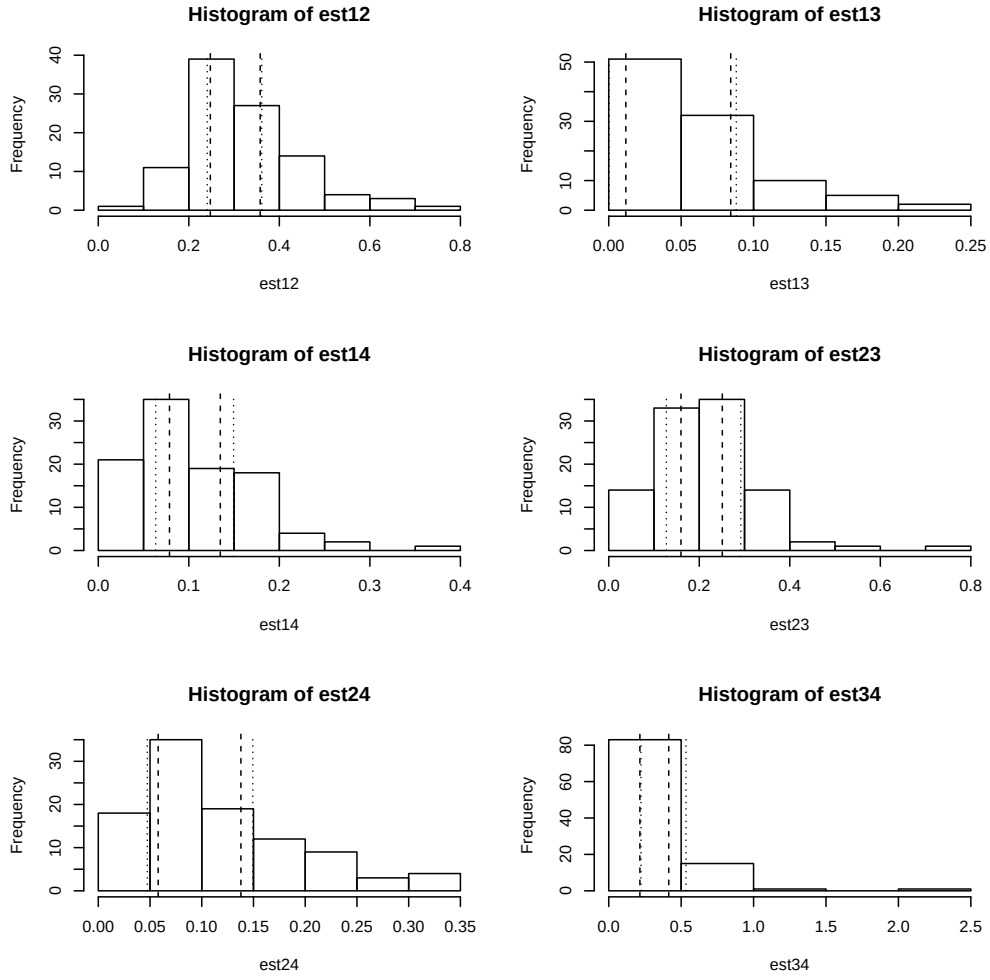
Programmet kjører her fint for datasettene med 500 og 200 individer, mens ved kjøring av datasettene med bare 20 individer får jeg tre advarsler. Advarslene har den



Figur 7.1: *Histogram fra 500 individer og 100 simuleringer.*



Figur 7.2: *Histogram fra 200 individer og 100 simuleringer.*



Figur 7.3: *Histogram fra 20 individer og 100 simuleringer, der de prikkete linjene er minimum og maksimum av estimatene generert med 500 individer, og de stiplete linjene er minimum og maksimum av estimatene generert med 200 individer.*

	500 individer	200 individer	20 individer
$\hat{q}_{12}$	0.30297	0.30097	0.32212
$\hat{q}_{13}$	0.05067	0.05082	0.05792
$\hat{q}_{14}$	0.10236	0.09940	0.10294
$\hat{q}_{23}$	0.20067	0.20370	0.21258
$\hat{q}_{24}$	0.09970	0.09809	0.11357
$\hat{q}_{34}$	0.30275	0.31217	0.34168

Tabell 7.1: *Gjennomsnittsverdiene av estimatene fra de tre kjøringene.*

	500 individer	200 individer	20 individer
$\hat{q}_{12}$	0.00040	0.00087	0.01525
$\hat{q}_{13}$	0.00013	0.00026	0.00277
$\hat{q}_{14}$	0.00013	0.00028	0.00452
$\hat{q}_{23}$	0.00037	0.00107	0.01255
$\hat{q}_{24}$	0.00022	0.00049	0.00617
$\hat{q}_{34}$	0.00116	0.00336	0.08170

Tabell 7.2: *MSE av estimatene.*

samme feilmeldingen som den jeg fikk i testingen i kapittel 6. Advarselen sier "at den ikke klarer å regne ut asymptotisk standardfeil, at Hessian ikke er positiv definit, og at optimaliseringen sannsynligvis ikke har konvergert til et maksimum likelihood". Jackson har i manualen[8] et helt avsnitt der han omtaler ulike årsaker for at konvergensen feiler. Han sier blant annet at optimaliseringen i noen omstendigheter kanskje vil rapportere konvergens, men feile med å beregne standardfeil. Hessian (matrise av andrederiverte) av log-likelihooden av den rapporterte løsningen er da i de tilfellene ikke positiv definit, som kanskje medfører at denne løsningen istedenfor er et sadelpunkt enn et maksimum likelihood, eller at den kan være nært et maksimum. Jackson tar også opp hvilke mulige endringer som kan gjøres for å unngå advarsler, men siden jeg her arbeider med simulerte datasett vil det være vanskelig å unngå dem helt. Ut fra andre kjøringer, enn de som er oppsummert i tabellene, ser det ikke ut til at advarsler forkommer så ofte.

I tilfellet her med 20 individer kan jeg, etter å ha studert de estimatene som har kommet ut, ikke se at de som har kommet fra advarslene skiller seg noe mer ut i forhold til de andre estimatene. Jeg velger derfor å se bort i fra disse advarslene her siden de ikke ser ut til å ha noen direkte innvirkning på resultatene når jeg kjører 100 simuleringer. Gjennomsnittverdiene som er gitt i tabell 7.1 er ganske nær de intensitetene som ble brukt for å generere datasettene. Dette viser at `msm` får verdier som ligger rundt de opprinnelige estimatene som ble gitt i intensitetsmatrisen $\mathbf{Q}$. Siden estimatene er mindre enn 1 medfører dette at MSE-verdiene som er i tabell 7.2 blir veldig små. Ut fra disse verdiene ser jeg at kjøringen med 500 individer gir de minste MSE-verdiene. Desto færre individer som er med i datasett fører til en økning av disse verdiene, der økningen er størst når en går fra 200 individer til 20

individer. Har også lagt merke til at kjøretiden ved disse kjøringene med 100 simuleringer varierer litt fra hverandre. Den krever betraktelig mer tid når datasettet består av 500 individer (ca. 20 min), og ved å redusere antall individer ned til 200 halveres kjøretiden, mens ved bare 20 individer krever den bare et par minutter på 100 simuleringer.

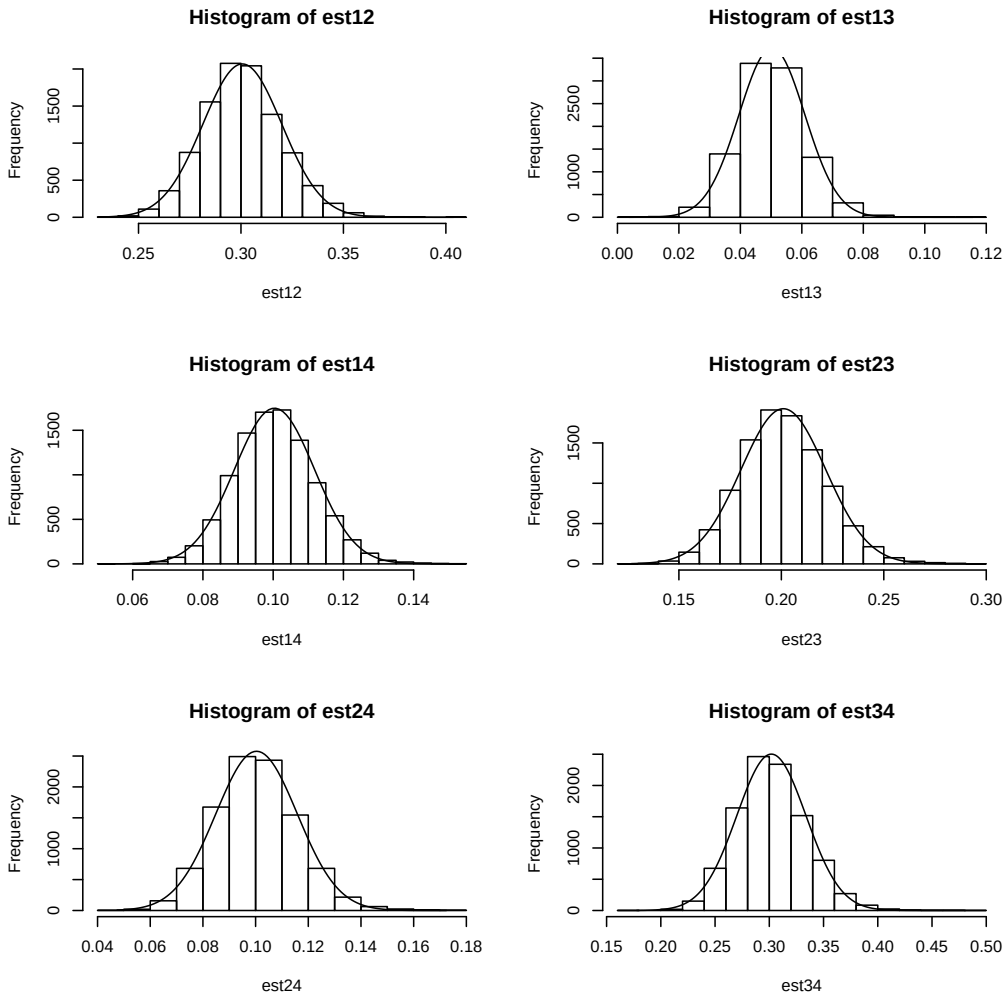
Histogrammene som er i figurene 7.1-7.3 viser alle de estimerte intensitetene. Formen på histogrammene minner litt om en normalfordeling der estimatene er sentrert rundt de gitte intensitetene i **Q**. Ved å sammenligne histogrammene ser en at de har forskjellige verdier på aksene som medfører at de kan se ganske like ut, hvis en ser bort fra at histogrammene for datasettene med 20 individer har noen ville estimater som får dem til å se litt asymmetriske ut. Har valgt å tegne inn vertikale linjer for minimums- og maksimumsverdiene fra de to andre kjøringene inn i histogrammene for 20 individer, se figur 7.3. Med disse linjene kommer forskjellene mellom kjøringene tydeligere fram. Ser at kjøringene med 500 og 200 individer får estimater som er nærmere de gitte intensitetene enn kjøringen med 20 individer. De ville verdiene som en får kan komme av at det ikke har vært mange nok observasjoner fra den ene tilstanden til den andre.

Resultatene fra kjøringene med 10 000 simuleringer er oppsummert i tabell 7.3 og i figurene 7.4-7.6. Kjøringen med 500 individer gir ingen advarsler, mens kjøringen med 200 individer gir 12 advarsler. Kjøringen med 20 individer gir 36 error som blir fjernet fra kjøringen, og 50 eller flere advarsler. Hvis en sammenligner de verdiene som er gitt i tabell 7.3 med de fra tabellene 7.1 og 7.2, ser det ikke ut fra disse tallene ut som at det er så stor forskjell mellom det å kjøre 100 eller 10 000 simuleringer, bortsett fra kjøretiden.

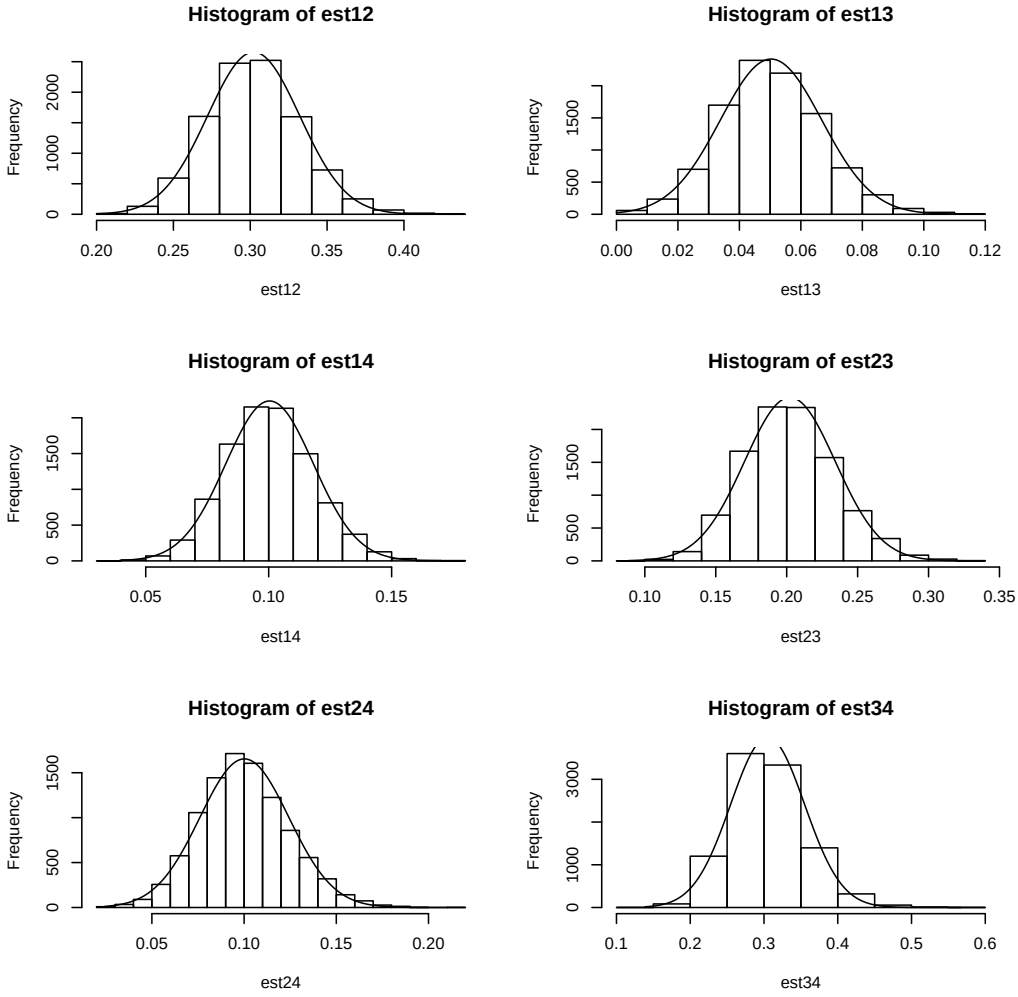
	500 individer		200 individer		20 individer	
	Gj.snitt	MSE	Gj.snitt	MSE	Gj.snitt	MSE
$\hat{q}_{12}$	0.30067	0.00037	0.30222	0.00091	0.31613	0.01069
$\hat{q}_{13}$	0.05027	0.00011	0.05030	0.00027	0.05570	0.00261
$\hat{q}_{14}$	0.10037	0.00013	0.10037	0.00032	0.10104	0.00348
$\hat{q}_{23}$	0.20098	0.00043	0.20227	0.00103	0.21578	0.01379
$\hat{q}_{24}$	0.10036	0.00024	0.10016	0.00058	0.10615	0.00756
$\hat{q}_{34}$	0.30193	0.00102	0.30491	0.00254	0.36182	0.16397

Tabell 7.3: *Tallene fra kjøringene med 10 000 simuleringer.*

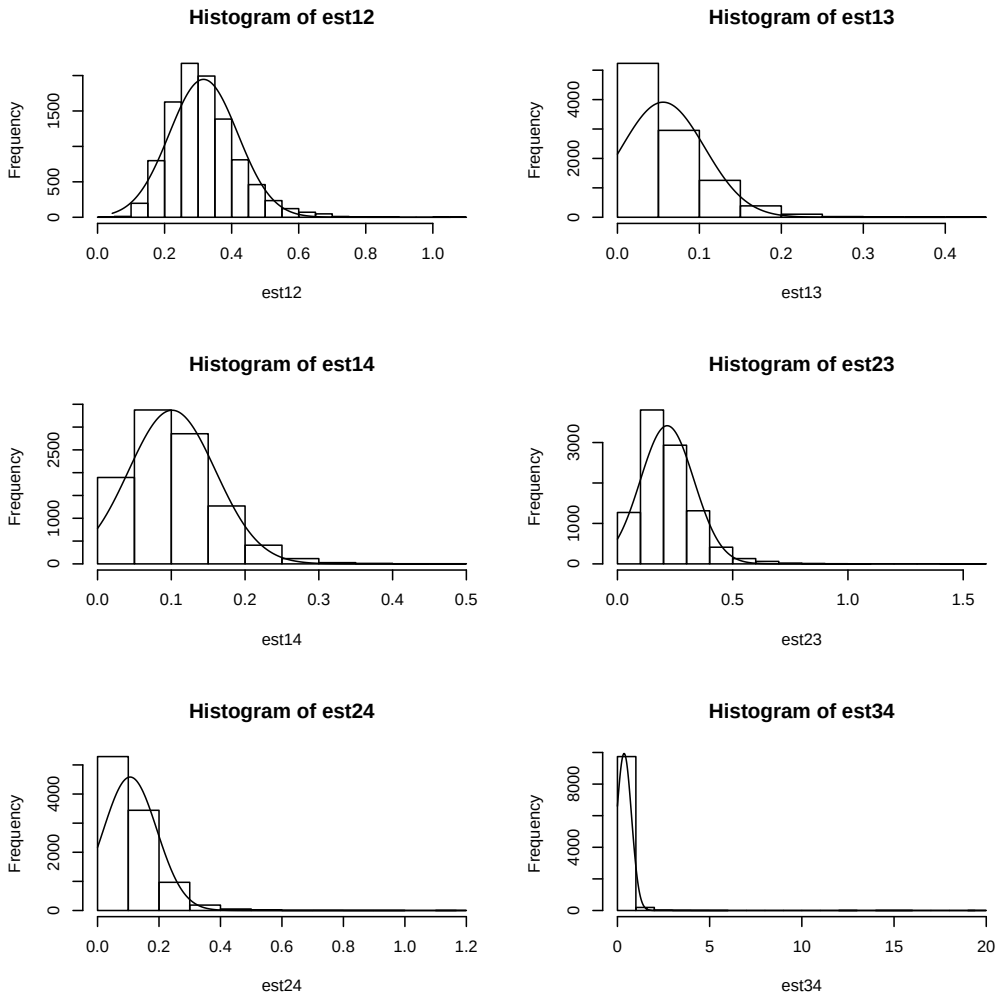
Formen på histogrammene i figurene 7.4 og 7.5 minner mye om en normalfordeling, og ved å tegne inn en glatt kurve over histogrammene, ved bruk av gjennomsnittsverdien og standardavviket av estimatene, ser tilpasningene ut til å være utmerket for alle de histogrammene. Ser også at der det i figur 7.1 og 7.2 kunne se ut til at de var asymmetriske, nå har forsvunnet ved å øke antall simuleringer. Histogrammene med normallinjene i figur 7.6 viser at fordelingen av estimatene ikke er normalfordelte. Ser at linjene ikke går høyt nok og det ser ut til å være en skjevhet i histogrammene. Det er bare histogrammet for $\hat{q}_{12}$-ene som ser ut til å være litt



Figur 7.4: *Histogram av gjennomsnittsverdiene av estimatene med 500 individer*



Figur 7.5: *Histogram av gjennomsnittsverdiene av estimatene med 200 individer*



Figur 7.6: *Histogram av gjennomsnittsverdiene av estimatene med 20 individer*

normalfordelt, noe som en kanskje kunne forvente siden det er flest overganger fra tilstand 1 til tilstand 2 i datasettene.

7.2.2 Oppsummering

Jeg kan ikke ut fra gjennomsnittsverdiene si noe om hvor mange individer som bør være med i datasettet og trenger derfor MSE-verdiene. Datasettene med 500 individer kommer best ut med de laveste MSE-ene. Forskjellene med å redusere antall individer ned til 200 er ikke så veldig store, bortsett fra at kjøretiden nesten blir halvert. Størst utslag er det når antall individer ble redusert helt ned til bare 20 individer. Histogrammene viser det samme som MSE-ene. 500 individer gir generelt de beste resultatene, mens 20 individer gir de dårligste resultatene og har kanskje litt for få individer. Datasettene med 200 individer gir nesten like gode resultater som de med 500 individer, og disse datasettene er i gjengjeld under halvparten så store. Dermed blir gevinsten for å samle inn over dobbelt så mye mer data kanskje ikke så stor i forhold til de resultatene i dette enkle tilfellet.

For videre arbeid med funksjonen `msm()` velger jeg å ta med 500 individer siden den gir de beste resultatene. Det trengs kanskje ikke så mange individer når jeg bare kjører den enkle modellen, men når jeg videre inkluderer kovariater kan det kanskje være lettere å få fram ønskede resultater om jeg tar med flest mulig individer.

Kapittel 8

Testing av metoden

I kapittel 7 ble robustheten til `msm` i forhold til hvor mange individer som bør være med i datasettet, testet. Det ble der tilpasset en multitilstandsmodell med en overgangsintensitetsmatrise $\mathbf{Q}$. I dette kapittelet skal jeg legge til ulike kovariater på multitilstandsmodellen og sjekke om funksjonen klarer å fange opp effekten av disse. Deretter skal jeg inkludere feil i datasettet ut fra en feilklassifiseringsmatrise $\mathbf{C}$, og bruke `msm` til å tilpasse en skjult Markovmodell.

8.1 Kovariater

8.1.1 Grunnlag

Jeg skal her modellere effektene av forklaringsvariabler på overgangsratene $q_{rs}(t)$ i tilfellet uten feilklassifisering. Lar intensitetsmatrisen avhenge av en kovariat vektor $\mathbf{z}$ der det første leddet antas å være lik 1. Intensiteten q_{rs} for et individ ved en observasjonstid t er da gitt som

$$q_{rs}(t, \mathbf{z}(t)) = \exp\{\mathbf{z}(t)\} = \exp\{\beta_{rs.0} + \beta_{rs.1} z_1(t) + \cdots + \beta_{rs.m} z_m(t)\}.$$

Her representerer $\exp\{\beta_{rs.0}\}$ overgangsintensiteten når den ikke er påvirket av noen kovariater. Er kovariatene tidsavhengige, antas den å være konstant mellom observasjonstidene av Markovprosessen. For å oppgi kovariater i `msm` kan en anvende argumentet `covariates`. Den vil da kalkulere sannsynlighetene for en overgang til en tilstand, fra tid t til tid u, ved å bruke kovariatverdiene ved tid t [8].

Jeg starter først med å bare inkludere en kovariat til modellen. Velger her at denne skal være en binær variabel med følgende kategorier: mann og kvinne. Regresjonslikningen er da gitt ved

$$q_{rs}(t, \mathbf{z}(t)) = \exp\{\beta_{rs.0} + \beta_{rs.1} z_1(t)\},$$

der $z_1(t)$ er en dummyvariabel for kjønn. Her er $z_1(t) = 0$ for kvinner og $z_1(t) = 1$ for menn. Kjønn endres ikke over tid, slik at $z_1(t) = z_1(t_1)$ for $t \geq 0$, der t_1 er den

første observasjonstiden. Får to intensitetsmatriser,

$$\mathbf{Q} = \begin{bmatrix} 0 & 0.3 & 0.05 & 0.1 \\ 0 & 0 & 0.2 & 0.1 \\ 0 & 0 & 0 & 0.3 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \qquad \mathbf{V} = \begin{bmatrix} 0 & 0.5 & 0.05 & 0.1 \\ 0 & 0 & 0.4 & 0.1 \\ 0 & 0 & 0 & 0.3 \\ 0 & 0 & 0 & 0 \end{bmatrix}.$$

Her er **Q**-matrisen den samme som i kapittel 7, og representerer her overgangsintensitetene for kvinner, mens **V**-matrisen representerer overgangsintensitetene for menn. **V** er valgt til å ha litt høyere verdier for q_{12} og q_{23}, mens de andre resterende intensitetene forblir de samme som i **Q**. I denne situasjonen betyr det at menn som er friske har en høyere sannsynlighet for å bli syke, og videre, etter de har blitt syke, en høyere sannsynlighet for å bli sykere. Jeg genererer her datasettet ved å spesifisere begge matrisene i kallet på `xMatrise`. Ved å sende inn to matriser forteller jeg funksjonen at den skal bruke to ulike matriser ut fra hvilket kjønn de er, se. vedlegg A.

Relative rate [5] for tilstedeværelse vs. fravær av eksponering (menn sammenlignet med kvinner) er

$$\frac{q_{rs}(t, z_1(t) = 1)}{q_{rs}(t, z_1(t) = 0)} = \frac{e^{\beta_{rs.0} + \beta_{rs.1}}}{e^{\beta_{rs.0}}} = e^{\beta_{rs.1}}.$$

I dette tilfellet er det bare to koeffisienter, $\beta_{12.1}$ og $\beta_{23.1}$, som er av interesse, siden de andre intensitetene er lik hverandre. Jeg får ut fra de valgte intensitetsmatrisene at:

$$e^{\beta_{12.1}} = \frac{0.5}{0.3} = 1.667 \qquad \Rightarrow \qquad \beta_{12.1} = 0.51083,$$

$$e^{\beta_{23.1}} = \frac{0.4}{0.2} = 2 \qquad \Rightarrow \qquad \beta_{23.1} = 0.69315.$$

Ser ut fra den relative raten for å gå fra tilstand 1 til 2 har en markant effekt på intensiteten med 1,67 for menn mot kvinner. Med en relative rate på 2 betyr det at intensiteten for å hoppe fra tilstand 2 til 3 er dobbelt så stor for menn enn for kvinner. Kjører her 100 simuleringer og lagrer de intensitetene som trengs for å regne ut β-ene. Jeg vil i resultatene komme tilbake til hvilke verdier det er.

Videre skal jeg også legge til en tidsavhengig kovariat i tillegg til den binære kovariaten. Alder kan her være en slik kovariat, der intensitetene avhenger av alderen på individene. Bestemmer her å inkludere alder, som skal være representert av tre aldersgrupper: gruppe 1, 2 og 3, henholdsvis 70-71 år, 72-73 år og 74-76 år. Regresjonslikningen for intensitetene blir da:

$$q_{rs}(t, \mathbf{z}(t)) = \exp\{\beta_{rs.0} + \beta_{rs.1} z_1(t) + \beta_{rs.2} z_2(t) + \beta_{rs.3} z_3(t)\}.$$

Der er $z_1(t)$ den samme som tidligere, mens $z_2(t)$ og $z_3(t)$ er dummyvariabler for aldersgruppene. For et individ som er i gruppe 1 (70-71 år) vil både $z_2(t) = 0$ og $z_3(t) = 0$. For et individ som er i gruppe 2 (72-73 år) vil $z_2(t) = 1$ og $z_3(t) = 0$, og intensiteten vil da være påvirket av koeffisienten $\beta_{rs.2}$. For et individ som er i

gruppe 3 (74-76 år) vil $z_2(t) = 0$ og $z_3(t) = 1$, og intensiteten vil da være påvirket av koeffisienten $\beta_{rs.3}$. I tillegg til matrisene **Q** og **V**, trenger jeg her fire matriser til. For menn velges de for å være

$$\mathbf{V2} = \begin{bmatrix} 0 & 0.7 & 0.05 & 0.1 \\ 0 & 0 & 0.6 & 0.1 \\ 0 & 0 & 0 & 0.3 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \qquad \mathbf{V3} = \begin{bmatrix} 0 & 0.9 & 0.05 & 0.1 \\ 0 & 0 & 0.8 & 0.1 \\ 0 & 0 & 0 & 0.3 \\ 0 & 0 & 0 & 0 \end{bmatrix},$$

der **V2** og **V3** representerer den økende effekten av alder for menn. Ønsker her at menn og kvinner i dette tilfellet skal påvirkes likt, så for å sette opp intensitetsmatrisene for kvinner trenger jeg å finne β-koeffisientene for aldergruppene ut fra matrisene gjeldende for menn. Jeg finner koeffisientene tilknyttet gruppe 2 ut fra

$$\frac{q_{rs}(t, z_1(t) = 1, z_2(t) = 1, z_3(t) = 0)}{q_{rs}(t, z_1(t) = 1, z_2(t) = 0, z_3(t) = 0)} = \frac{e^{\beta_{rs.0}+\beta_{rs.1}+\beta_{rs.2}}}{e^{\beta_{rs.0}+\beta_{rs.1}}} = e^{\beta_{rs.2}},$$

som gir

$$e^{\beta_{12.2}} = \frac{0.7}{0.5} = 1.4 \qquad \Rightarrow \qquad \beta_{12.2} = 0.33647,$$
$$e^{\beta_{23.2}} = \frac{0.6}{0.4} = 1.5 \qquad \Rightarrow \qquad \beta_{23.2} = 0.40547.$$

For gruppe 3 blir koeffisientene funnet ut fra

$$\frac{q_{rs}(t, z_1(t) = 1, z_2(t) = 0, z_3(t) = 1)}{q_{rs}(t, z_1(t) = 1, z_2(t) = 0, z_3(t) = 0)} = \frac{e^{\beta_{rs.0}+\beta_{rs.1}+\beta_{rs.3}}}{e^{\beta_{rs.0}+\beta_{rs.1}}} = e^{\beta_{rs.3}},$$

som gir

$$e^{\beta_{12.3}} = \frac{0.9}{0.5} = 1.8 \qquad \Rightarrow \qquad \beta_{12.3} = 0.58779,$$
$$e^{\beta_{23.3}} = \frac{0.8}{0.4} = 2 \qquad \Rightarrow \qquad \beta_{23.3} = 0.69315.$$

Jeg kan nå med disse koeffisientene finne intensitetene for kvinner

$$q_{12}(t, z_1(t) = 0, z_2(t) = 1, z_3(t) = 0) = e^{\beta_{12.0}+\beta_{12.2}} = e^{-1.20397+0.33647} = 0.42$$
$$q_{23}(t, z_1(t) = 0, z_2(t) = 1, z_3(t) = 0) = e^{\beta_{23.0}+\beta_{23.2}} = e^{-1.60944+0.40547} = 0.3$$
$$q_{12}(t, z_1(t) = 0, z_2(t) = 0, z_3(t) = 1) = e^{\beta_{12.0}+\beta_{12.3}} = e^{-1.20397+0.58779} = 0.54$$
$$q_{23}(t, z_1(t) = 0, z_2(t) = 0, z_3(t) = 1) = e^{\beta_{23.0}+\beta_{23.3}} = e^{-1.60944+0.69315} = 0.4$$

og med de andre intensitetene uendret vil det gi matrisene

$$\mathbf{Q2} = \begin{bmatrix} 0 & 0.42 & 0.05 & 0.1 \\ 0 & 0 & 0.3 & 0.1 \\ 0 & 0 & 0 & 0.3 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \qquad \mathbf{Q3} = \begin{bmatrix} 0 & 0.54 & 0.05 & 0.1 \\ 0 & 0 & 0.4 & 0.1 \\ 0 & 0 & 0 & 0.3 \\ 0 & 0 & 0 & 0 \end{bmatrix},$$

der **Q2** og **Q3** representerer den økende effekten av alder for kvinner. Intensitetene for at et individ som er frisk skal bli syk, og at en som er syk skal bli sykere, øker når individet kommer i en ny aldersgruppe.

```
Call:
msm(formula = tilstand ~ tid, subject = ptnum, data = datafil,
     qmatrix = qmatrise, death = 4)

Maximum likelihood estimates:
Transition intensity matrix

        State 1                    State 2
State 1 -0.5327 (-0.5831,-0.4867) 0.3868 (0.3424,0.4369)
State 2 0                          -0.4026 (-0.4595,-0.3528)
State 3 0                          0
State 4 0                          0
        State 3                   State 4
State 1 0.06755 (0.04525,0.1009) 0.07841 (0.05892,0.1043)
State 2 0.2818 (0.2372,0.3348)   0.1209 (0.09165,0.1594)
State 3 -0.285 (-0.3389,-0.2397) 0.285 (0.2397,0.3389)
State 4 0                         0

-2 * log-likelihood:  4207.752
```

Tabell 8.1: Utskrift av kjøring der kovariaten ikke er spesifisert i `msm`.

8.1.2 Resultat

Tabellene 8.1 og 8.2 viser eksempler på utskrifter av `msm`-objektet, først uten å spesifisere kovariaten for kjønn, og en der den er blitt spesifisert i kallet på `msm`. Tabell 8.1 viser maksimum likelihood estimatene og de tilhørende 95 % konfidensintervallene. Ut fra den estimerte intensitetsmatrisen kan jeg se at individer har ca. 5 ganger så høyere sannsynlighet for å utvikle en sykdom enn å dø frisk. Den viser også at et individ som er blitt syk har omtrent dobbelt så stor sannsynlighet for å bli sykere enn å dø, og for en som er blitt sykere er sjansen stor for en snarlig død. Tabell 8.2 inkluderer også de estimerte kovariateffektene av $z_1(t) = 1$ (menn) og de tilhørende konfidensintervallene. De effektene som vises her er altså $\beta_{rs.1}$-verdiene. Utskriften viser i dette tilfellet, ut fra de log-lineære effektene av kjønn, at tre av de er signifikante. Ved å studere konfidensintervallene ser det ut til at signifikansen er litt høyere for tilstand 1 til tilstand 2 og for tilstand 2 til tilstand 3. Siden effektene er positive kan det tolkes som at menn (som her er den variabelen som er lik 1) har en høyere sykdomsutvikling enn for kvinner, som er hva en kunne forvente seg fra de valgte intensitetsmatrisene $\mathbf{Q}$ og $\mathbf{V}$.

Utskriften i tabell 8.2 viser overgangsintensitetsmatrisen der kovariatene er satt til deres gjennomsnitt. For å få separate intensitetsmatriser for kvinner og menn, kan man bruke en funksjon som også er inneholdt i pakken msm, `qmatrix.msm()`. Denne tar blant annet inn objektet fra `msm` og en `covariates` spesifikasjon, som enten vil være `kjonn==0` for kvinner eller `kjonn==1` for menn. Tabell 8.3 viser, her fra 100 simuleringer, gjennomsnittsverdiene og MSE-ene av de intensitetene som kommer fra

```
Call:
msm(formula = tilstand ~ tid, subject = ptnum, data = datafil,
     qmatrix = qmatrise, covariates = ~kjonn, death = 4)

Maximum likelihood estimates:
Transition intensity matrix with covariates set to their means

        State 1                    State 2
State 1 -0.5354 (-0.5864,-0.4889) 0.3911 (0.3461,0.442)
State 2 0                          -0.3965 (-0.4536,-0.3465)
State 3 0                          0
State 4 0                          0
        State 3                    State 4
State 1 0.06507 (0.0429,0.0987)   0.07927 (0.0596,0.1054)
State 2 0.2778 (0.2333,0.3308)    0.1187 (0.08939,0.1575)
State 3 -0.2666 (-0.3213,-0.2213) 0.2666 (0.2213,0.3213)
State 4 0                          0

Log-linear effects of kjonn

        State 1 State 2               State 3
State 1 0       0.3963 (0.1506,0.642) 0.5684 (-0.2694,1.406)
State 2 0       0                     0.5195 (0.1731,0.8659)
State 3 0       0                     0
State 4 0       0                     0
        State 4
State 1 0.333 (-0.2396,0.9056)
State 2 -0.01475 (-0.5914,0.5619)
State 3 0.4414 (0.07802,0.8048)
State 4 0

-2 * log-likelihood:  4172.313
```

Tabell 8.2: *Utskrift fra kjøring, der kovariaten er spesifisert.*

	Q	Gj.snitt	MSE	**V**	Gj.snitt	MSE
$\hat{q}_{12}$	0.3	0.30008	0.00072	0.5	0.49198	0.00187
$\hat{q}_{13}$	0.05	0.04755	0.00013	0.05	0.06026	0.00060
$\hat{q}_{14}$	0.1	0.10028	0.00023	0.1	0.10357	0.00047
$\hat{q}_{23}$	0.2	0.20141	0.00077	0.4	0.39349	0.00184
$\hat{q}_{24}$	0.1	0.09789	0.00037	0.1	0.10578	0.00079
$\hat{q}_{34}$	0.3	0.29743	0.00210	0.3	0.29874	0.00136

Tabell 8.3: *Resultatene over verdiene hentet ut fra* `qmatrix` *fra simuleringene med en kovariat.*

`qmatrix.msm`. Tabellen viser her at gjennomsnittsverdiene er nært de valgte verdiene i **Q** og **V**, og med lave MSE-verdier kan det virke som at `msm` klarer å fange opp de forskjellene som er i matrisene. $\hat{\beta}$-ene blir regnet ut ved å ta logaritmen enten av $\hat{q}_{rs}(t, z_1(t) = 0)$, eller av $\hat{q}_{rs}(t, z_1(t) = 1)/\hat{q}_{rs}(t, z_1(t) = 0)$, der $(r, s) \in \{(1, 2), (2, 3)\}$. Tabell 8.4 viser gjennomsnittsverdiene og MSE-ene av de estimerte β-verdiene målt opp mot de opprinnelige β-koeffisientene. Ser her at den klarer å estimere de koeffisientene som jeg ønsket rimelig bra.

	β	Gj.snitt	MSE
$\hat{\beta}_{12.0}$	-1.20397	-1.20770	0.00799
$\hat{\beta}_{23.0}$	-1.60944	-1.61178	0.01881
$\hat{\beta}_{12.1}$	0.51083	0.49468	0.01545
$\hat{\beta}_{23.1}$	0.69315	0.67317	0.03344

Tabell 8.4: *Verdiene på β-koeffisientene.*

Følgende utskrift viser et eksempel på et `msm`-objekt når det også legges til flere kovariater i kallet på `msm`.

```
Call:
msm(formula = tilstand ~ tid, subject = ptnum, data = datafil,
     qmatrix = qmatrise, covariates = ~kjonn + gruppe2 + gruppe3,
     death = 4)

Maximum likelihood estimates:
Transition intensity matrix with covariates set to  their means

        State 1                    State 2
State 1 -0.4967 (-0.5613,-0.4395) 0.3357 (0.2836,0.3973)
State 2 0                         -0.406 (-0.471,-0.35)
State 3 0                         0
State 4 0                         0
        State 3                   State 4
```

```
State 1 0.06857 (0.04257,0.1104)  0.09245 (0.06495,0.1316)
State 2 0.3153 (0.2622,0.3791)    0.0907 (0.05992,0.1373)
State 3 -0.3438 (-0.4261,-0.2775) 0.3438 (0.2775,0.4261)
State 4 0                         0

Log-linear effects of kjonn

        State 1 State 2                 State 3
State 1 0       0.3311 (0.08161,0.5805) 0.04546 (-0.8153,0.9063)
State 2 0       0                       0.4627 (0.1227,0.8028)
State 3 0       0                       0
State 4 0       0                       0
        State 4
State 1 0.1746 (-0.312,0.6611)
State 2 -0.2324 (-0.9679,0.5031)
State 3 0.02063 (-0.319,0.3602)
State 4 0

Log-linear effects of gruppe2

        State 1 State 2                  State 3
State 1 0       -0.1517 (-0.4861,0.1827) 0.115 (-0.8697,1.1)
State 2 0       0                        0.2295 (-0.1692,0.6282)
State 3 0       0                        0
State 4 0       0                        0
        State 4
State 1 -0.123 (-0.7593,0.5134)
State 2 0.3992 (-0.4565,1.255)
State 3 -0.1925 (-0.701,0.316)
State 4 0

Log-linear effects of gruppe3

        State 1 State 2                  State 3
State 1 0       -0.2307 (-0.7386,0.2773) 0.2492 (-1.015,1.513)
State 2 0       0                        0.259 (-0.19,0.7079)
State 3 0       0                        0
State 4 0       0                        0
        State 4
State 1 -0.297 (-1.404,0.81)
State 2 0.1067 (-0.8991,1.112)
State 3 0.04318 (-0.436,0.5224)
State 4 0

-2 * log-likelihood:  4165.225
```

Utskriften tar nå også med de estimerte effektene og de tilhørende konfidensintervallene til de kovariatene som er blitt inkludert. Denne viser at ingen av alderseffektene er signifikante, og at to effekter av kjønn er signifikante. Dersom jeg kjører `msm` på det samme simulerte datasettet, men velger å sende inn andre initialverdier i `qmatrix`-argumentet, får jeg litt andre tall enn den ovenfor, som følgende utskrift viser

```
Call:
msm(formula = tilstand ~ tid, subject = ptnum, data = datafil,
     qmatrix = qmatrise, covariates = ~kjonn + gruppe2 + gruppe3,
     death = 4)

Maximum likelihood estimates:
Transition intensity matrix with covariates set to  their means

        State 1                    State 2
State 1 -0.5438 (-0.6126,-0.4827) 0.3599 (0.3075,0.4212)
State 2 0                         -0.4234 (-0.4894,-0.3664)
State 3 0                         0
State 4 0                         0
        State 3                    State 4
State 1 0.116 (0.08536,0.1577)    0.06785 (0.03649,0.1261)
State 2 0.3119 (0.2627,0.3702)    0.1116 (0.07331,0.1698)
State 3 -0.3575 (-0.4662,-0.2742) 0.3575 (0.2742,0.4662)
State 4 0                         0

Log-linear effects of kjonn

        State 1 State 2                State 3
State 1 0       0.3307 (0.08637,0.575) 0.3436 (-0.1915,0.8787)
State 2 0       0                      0.4782 (0.1575,0.7989)
State 3 0       0                      0
State 4 0       0                      0
        State 4
State 1 0.2747 (-0.3711,0.9205)
State 2 -0.01541 (-0.5871,0.5563)
State 3 -0.04703 (-0.3732,0.2791)
State 4 0

Log-linear effects of gruppe2

        State 1 State 2               State 3
State 1 0       0.418 (0.1367,0.6994) -0.1918 (-0.9498,0.5662)
State 2 0       0                     0.1601 (-0.2,0.5201)
State 3 0       0                     0
State 4 0       0                     0
        State 4
State 1 0.4774 (-0.2079,1.163)
```

```
State 2 0.806 (-0.1054,1.717)
State 3 -0.05173 (-0.6635,0.56)
State 4 0

Log-linear effects of gruppe3

        State 1 State 2                    State 3
State 1 0       -0.01422 (-0.5064,0.4779) 0.5326 (-0.2571,1.322)
State 2 0       0                         -0.01412 (-0.4527,0.4244)
State 3 0       0                         0
State 4 0       0                         0
        State 4
State 1 -0.4495 (-2.628,1.729)
State 2 0.5688 (-0.4209,1.558)
State 3 0.2219 (-0.369,0.8129)
State 4 0

-2 * log-likelihood:  4211.045
```

Jeg ser her, med en annen initialmatrise, at den ene effekten hos gruppe2 nå er blitt signifikant. Dette betyr for denne studien at sjansen for å gå fra tilstand 2 til tilstand 3 er høyere for personer som er 74-76 år. Ved å gjøre flere små endringer bare i initialmatrisen får jeg hver gang litt forskjellige resultater, og ulike signifikante effekter.

For å se nærmere på prosessen av optimaliseringsalgoritmen kan en spesifisere et `control`-argument til `msm`, som vil gå internt til `optim`-funksjonen [8]. Velger her å bruke `control = list(trace=1, REPORT=1)` i kallet. Det resulterer i at jeg får en lang liste som viser gangen i Nelder-Mead optimaliseringen. Helt i begynnelsen av listen gir den informasjon om funksjonsverdien (-2*log-likelihood) for initialparameterne. De starter her med forskjellige tall, der den første kjøringen har verdien 4234,368, mens den andre kjøringen har verdien 4691,217. Ser også at den avslutter ved bruk av 501 funksjonsvurderinger. Dette kan kanskje være en årsak til at `msm` gir litt forskjellige resultater.

Jeg prøver videre å sjekke om det hjelper å bruke en annen metode som for eksempel BFGS. Ved å anvende denne optimaliseringsalgoritmen ser jeg at den arbeider en del raskere enn Nelder-Mead, og bruker en del færre iterasjoner. De starter med de samme initialverdiene som ved Nelder-Mead, men begge ender her med det samme tallet 4161.7234, der den første trengte 32 iterasjoner og den andre trengte 34 iterasjoner. Utskriftene med denne metoden er nesten identiske.

8.1.3 Oppsummering

Ut fra kjøringen med bare en kovariat, ser det ikke ut til at `msm` har problemer med å klare å fange opp de effektene som er lagt til i datasettene. Derimot, når jeg også inkluderer en tidsavhengig kovariat, legger jeg merke til at bare valget av

initialmatrisen medfører at `msm` gir forskjellige resultater. Men problemet ser ut til å bli løst ved å bruke en annen metode enn defaulten.

8.2 Feilklassifisering

8.2.1 Grunnlag

Msm-pakken er laget for å kunne modellere feilklassifiseringer i et datasett. Jeg skal her prøve å tilpasse en skjult Markovmodell med feilklassifiseringsmatrisen **C** på et datasett som inneholder observasjoner som strider imot den underliggende modellen med overgangsintensitetsmatrisen **Q**. For å få et datasett som inneholder feil, genererer jeg først et datasett uten feil, ved hjelp av funksjonene: `xMatrise` og `zMatrise`. Deretter sendes objektet fra `zMatrise` inn i en ny funksjon, som jeg har valgt å kalle `feilMatrise`, se vedlegg A. Denne tar inn et datasett og en feilklassifiseringsmatrise **C**, som her er valgt til å være:

$$\mathbf{C} = \begin{bmatrix} 0.95 & 0.05 & 0 & 0 \\ 0.05 & 0.85 & 0.1 & 0 \\ 0 & 0.1 & 0.9 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

der rekkene summeres til 1. Denne funksjonen vil da gå igjennom alle observasjonene i datasettet og tilføre feile observasjoner ut fra **C**-matrisen. Jeg simulerer her feilklassifiseringer med **C** og bruker denne for å sammenligne resultatene fra `msm`. Med det nye datasettet kan jeg nå definere en `ematrix` i kallet på `msm`. Velger her i tillegg å bruke BFGS-metoden, som jeg har fra kapittel 6, skulle være en raskere metode.

8.2.2 Resultat

Tabellene 8.6 og 8.7 viser resultatene fra en kjøring. For å oppsummere multitilstandsdata kan en bruke funksjonen `statetable.msm()`, som produserer en frekvenstabell for par av påfølgende tilstander [8]. En slik frekvenstabell for dette tilfellet er gitt i tabell 8.5. Jeg ser her at det er 57 observasjoner som går fra tilstand 2 til tilstand 1, 2 observasjoner som går fra tilstand 3 til tilstand 1, mens det er 77 observasjoner som går fra tilstand 3 til tilstand 2.

	to			
from	1	2	3	4
1	749	290	108	113
2	57	365	176	81
3	2	77	257	120

Tabell 8.5: *Oppsummering av alle overgangene i datasettet.*

Feilklassifiseringsmatrisen, som er gitt i tabell 8.6, viser det er blitt estimert en

```
> eks.msm

Call:
msm(formula = tilstand ~ tid, subject = ptnum, data = datasett,
     qmatrix = qmatrise, ematrix = ematrise, death = 4, method = "BFGS")

Maximum likelihood estimates:
Transition intensity matrix

        State 1                   State 2
State 1 -0.4699 (-0.5159,-0.428) 0.3334 (0.2877,0.3862)
State 2 0                         -0.3551 (-0.4183,-0.3015)
State 3 0                         0
State 4 0                         0
        State 3                    State 4
State 1 0.05448 (0.0296,0.1003)    0.08207 (0.06287,0.1071)
State 2 0.2596 (0.2048,0.3291)     0.09549 (0.06346,0.1437)
State 3 -0.3359 (-0.4074,-0.2769) 0.3359 (0.2769,0.4074)
State 4 0                          0

Misclassification matrix

        State 1                    State 2
State 1 0.9477 (0.9304,0.9608)     0.05235 (0.03918,0.06962)
State 2 0.05082 (0.03085,0.08262) 0.8394 (0.7415,0.905)
State 3 0                          0.1084 (0.06706,0.1706)
State 4 0                          0
        State 3                  State 4
State 1 0                        0
State 2 0.1098 (0.07404,0.1597) 0
State 3 0.8916 (0.8294,0.9329)  0
State 4 0                        1 (1,1)

-2 * log-likelihood:  5280.419
```

Tabell 8.6: *Utskrift fra kjøring med feilklassifisering.*

sannsynlighet på ca. 0,05 for at tilstand 1 vil bli diagnosert feil til å være i tilstand 2, og tilsvarende for at tilstand 2 vil bli feildiagnosert til å være i tilstand 1. Ser videre at sannsynligheten for å feilklassifisere tilstand 2 til å være tilstand 3, og motsatt vei, er litt høyere med ca. 0,11. Ved å sammenligne disse estimatene med de som ble brukt for å lage feil i observasjonene, kan en se at `msm` har klart å estimere disse parameterne rimelig godt. Dette kommer også frem i tabell 8.7, når jeg kjører 100 simuleringer. Tabellen viser gjennomsnittsverdiene av de estimerte parameterne med de tilhørende MSE-verdiene. Legger her merke til at MSE-verdiene for $\hat{c}_{12}$ og $\hat{c}_{23}$, altså estimatene for å klassifisere tilstandene feil oppover, er lavere enn MSE-verdiene for $\hat{c}_{21}$ og $\hat{c}_{32}$, som er estimatene for å klassifisere tilstandene feil nedover.

	Gj.snitt	MSE		Gj.snitt	MSE
$\hat{q}_{12}$	0.30631	0.00057	$\hat{c}_{12}$	0.04999	0.00004
$\hat{q}_{13}$	0.04950	0.00021	$\hat{c}_{21}$	0.05043	0.00270
$\hat{q}_{14}$	0.09940	0.00014	$\hat{c}_{23}$	0.10180	0.00030
$\hat{q}_{23}$	0.19855	0.00060	$\hat{c}_{32}$	0.09639	0.00999
$\hat{q}_{24}$	0.10269	0.00027			
$\hat{q}_{34}$	0.30363	0.00131			

Tabell 8.7: *Resultatene fra 100 simuleringer.*

Dersom `msm` kjøres uten at `ematrix` blir spesifisert i kallet, vil den for dette genererte datasettet avbryte kjøringen og komme med en advarselsbeskjed som sier at: "datasettet kan være uforenlig med overgangsmatrisen for modellen uten feilklassifisering: individ 1 beveger seg fra tilstand 2 til tilstand 3 ved overgang 5". Det er altså ikke mulig å kjøre feil modell på datasettet, og hvis det skjer får jeg en klar beskjed om hva som er årsaken bak erroren. For å sjekke hvilken modell som passer for datasettet, dersom det i forhold til overgangsintensitetsmatrisen som ligger til grunn, trengs å bli spesifisert en feilklassifiseringsmatrise, vil funksjonen `statetable.msm` gi en enkel oversikt om det har forekommet noen hopp mellom tilstander som ikke er tillatt.

8.2.3 Oppsummering

Ut fra simuleringene ser det ut til at `msm` fungerer rimelig bra på de genererte datasettene. Den klarer å fange opp rimelige overgangsintensiteter og klassifiseringssannsynligheter i forhold til de som ble brukt til å produsere datasettene.

8.3 Utprøving av ulike klassifiseringssannsynligheter

Jeg har nå sett fra kjøringen med feilklassifiseringer at `msm` ikke har hatt noen problemer med å estimere de ulike parameterne. Ønsker derfor videre å sjekke hva som skjer dersom det blir for mange gale observasjoner i det genererte datasettet. Jeg vil se om det er en grense for hvor mange feil som kan være med i datasettet før estimeringen går galt og den ikke lenger klarer å fange opp rimelige estimater.

	$\hat{q}_{12}$	$\hat{q}_{13}$	$\hat{q}_{14}$	$\hat{q}_{23}$	$\hat{q}_{24}$	$\hat{q}_{34}$	
Valgte	0.3	0.05	0.1	0.2	0.1	0.3	
1.	0.33339	0.05450	0.08208	0.25961	0.09548	0.33584	
2.	0.30905	0.05925	0.09192	0.23795	0.09064	0.33063	
3.	0.34811	0.06519	0.08273	0.22726	0.11035	0.30200	
4.	0.34175	0.06540	0.06717	0.19618	0.13562	0.32196	
5.	0.42734	0.00002	0.08365	0.44320	0.08080	0.29885	
6.	0.36030	0.4e-05	0.07529	0.14267	0.14844	0.45137	
7.	-	-	-	-	-	-	**
8.	7.85950	0.00003	0.00006	0.18844	0.09605	0.24858	*
9.	16277.4	0.00138	0.02374	0.19630	0.08065	0.26973	*
10.	0.39727	0.02567	0.07524	0.39163	0.12666	0.25421	
11.	107068	0.00169	0.00360	0.18307	0.07975	0.29021	*
12.	0.42726	0.00150	0.08124	0.33323	0.12401	0.27079	

Tabell 8.8: *De estimerte intensitetene, der * betyr advarsel og ** betyr error.*

	$c_{12}/\hat{c}_{12}$	$c_{21}/\hat{c}_{21}$	$c_{23}/\hat{c}_{23}$	$c_{32}/\hat{c}_{32}$	
1.	0.05/0.05235	0.05/0.05082	0.1/0.10973	0.1/0.10843	
2.	0.2/0.21127	0.1/0.10776	0.1/0.10935	0.2/0.20796	
3.	0.2/0.18467	0.2/0.23748	0.2/0.16510	0.2/0.19238	
4.	0.3/0.28418	0.2/0.19932	0.2/0.19571	0.3/0.26606	
5.	0.4/0.40606	0.2/0.30420	0.2/0.15218	0.4/0.48143	
6.	0.5/0.47113	0.2/0.12415	0.2/0.25053	0.5/0.32879	
7.	0.5/ -	0.3/ -	0.3/ -	0.5/ -	**
8.	0.5/0.52598	0.25/0.43369	0.25/0.06258	0.5/0.49929	*
9.	0.5/0.60201	0.2/0.36327	0.2/0.05613	0.6/0.60034	*
10.	0.4/0.39439	0.2/0.29161	0.2/0.18131	0.6/0.61790	
11.	0.6/0.60000	0.2/0.33632	0.2/0.05334	0.4/0.50043	*
12.	0.3/0.26035	0.2/0.21894	0.2/0.17308	0.7/0.72147	

Tabell 8.9: *Viser de valgte klassifiseringssannsynlighetene mot de estimerte, der * betyr advarsel og ** betyr error.*

8.3.1 Resultat

Tabellene 8.8 og 8.9 viser resultatene fra kjøringene, der den første viser de estimerte intensitetene mot den valgte intensiteten og den andre viser de estimerte klassifiseringssannsynlighetene mot de valgte parameterene. Jeg kan fra tabell 8.8 se, ut fra av de tilfellene som ikke har gitt en advarsel, at den ikke helt klarer å estimere de underliggende intensitetene like godt når datasettene er inneholdt en del feil. Legger spesielt merke til ved kjøring nr. 5, 6 og 12 at den ikke klarer å estimere q_{13}, slik at den er blitt veldig liten. Kan ut fra tabell 8.9, dersom en ser bort fra de som har fått en advarsel, se at den har klart å estimere c_{12} rimelig bra i forhold til de valgte estimatene. Ser for de andre klassifiseringssannsynlighetene, at `msm` ikke helt klarer å fange opp alle estimatene like godt når jeg har økt sannsynlighetene for feil.

8.3.2 Oppsummering

Ser at hvis man har for mange feilklassifiseringer i datasettet vil ikke `msm` klare å fange opp alle de ulike effektene, like bra. Men `msm` ser ut til å fungere bra for å tilpasse en skjult Markovmodell i de situasjonene der klassifiseringssannsynlighetene ikke er for store.

Kapittel 9

Konklusjon og videre arbeid

9.1 Konklusjon

Hovedfokuset i denne oppgaven har vært å sette opp en aktuell modell for sykdomsprogresjon, og sjekke om msm-pakken fungerer godt til å modellere denne typen problemstillinger. Etter å ha satt opp uttrykket for likelihooden til den tidskontinuerlige Markovmodellen, først uten, og til slutt med feilklassifisering, har jeg sett at de kan bli ganske store og kompliserte til å gjøre videre beregninger på, spesielt for den skjulte Markovmodellen. Jeg har ut fra dette konkludert med at for å estimere parameterene, trenger jeg en numerisk metode for å løse likelihooden.

Jeg har studert msm-pakken på mine egne simulerte datasett og sett at `msm` har kjørt fint i de fleste tilfellene. Den klarer å estimere rimelige parametere ut fra hvordan jeg har generert datasettene mine på. Når jeg inkluderte den tidsavhengige kovariaten kunne jeg se at valg av optimaliseringsmetode var av betydning i forhold til å få omtrent de samme resultatene, uansett hvilke initialverdier som ble sendt inn i `qmatrix`. Nelder-Mead-metoden ga ulike resultater når jeg endret på initialverdiene, mens BFGS-metoden ga de samme resultatene. Det å tilpasse modellen for feilklassifiseringer ga ingen problemer, bortsett fra når jeg genererte datasett med mange feilklassifiseringer.

Konklusjonen ut fra utprøvingen av msm-pakken er at den ser ut til å fungere rimelig godt dersom man bruker de rette argumentene. Det å tilpasse en tidkontinuerlig- og en skjult Markovmodell på denne typen data virker som et godt valg av modell. Den ser ut til å beskrive datasettet ganske bra.

9.2 Videre arbeid

Hadde jeg hatt mer tid til rådighet kunne det ha vært interessant å se nærmere på de andre funksjonene som også finnes i msm-pakken. I prinsippet kan resultatene fra `msm` bli brukt til testing, eksempelvis i likelihood-ratio-testen. Msm-pakken inneholder en del flere funksjoner, blant annet `lrtest.msm()`. Denne bruker resultatene fra to eller flere tilpassede multitilstandsmodeller fra `msm`, der de må være tilpasset på det samme datasettet [7].

Et forslag til videre arbeid kan være å sammenligne denne modellen mot andre metoder som er blitt anvendt for å finne de aktuelle parameterne til denne typen datasett. Det er for eksempel blitt brukt en Kaplan-Meier estimator for å finne estimatorene i tilsvarende situasjoner der de hopper mellom ulike tilstander i kontinuerlig tid [13]. Det kunne vært interessant å tilpasse en skjult Markovmodell på datasett der Kaplan-Meier estimatoren er brukt, og sammenligne de resultatene med de estimatene som kommer ut av `msm`, og teste ut hvilke av de to som beskriver datasettet best.

Det brukes diverse metoder i søket på å finne gode estimater som beskriver datasett med observasjoner i kontinuerlig tid. Hensikten med å studere multitilstandsmodellen med og uten feilklassifiseringer, og med den tilhørende pakken som jeg har arbeidet med i oppgaven, er å finne en modell som er bedre tilpasset datasettet. Siden msm-pakken ser ut til å passe rimelig bra på tilfellene jeg har studert, vil dette muligens være en modell som vil bli mer brukt i fremtiden.

Tillegg A

Programkode

A.1 Generering av datasett

```
xMatrise = function(ant,Q,V=NULL,Q2=NULL,Q3=NULL,V2=NULL,V3=NULL){
# Tar inn antall individer og en intensitetsmatrise Q. Den har
# også mulighet til å ta inn flere matriser. Ved å spesifisere
# både Q og V vil funksjonen generere et datasett ut fra en
# kovariat. Ved å spesifisere de andre matrisene vil funksjonen
# generere et datasett med en tidsavhengig kovariat.
# Returnerer en matrise med informasjon om tilstandene og
# overgangstidene til alle individene.

  xFunksjon = function(P,i,j,P2=NULL,P3=NULL){
  # Tar inn en overgangsmatrise, identifikasjonsnummer, et
  # tall som forteller hvilken rekke den er i matrisen, og
  # med mulighet for å spesifisere matriser for en tidsavhengig
  # kovariat. Returnerer informasjon for et individ.

    # Første kolonne
    A = matrix(data=0, nrow=2, ncol=7, dimnames = list(c(j,(j+1)),
                          c("ptnum", "alder", "tid", "tilstand",
                          "kjonn", "aldergr2", "aldergr3")))
    A[1] = i
    A[1,2] = alder
    A[1,3] = tid
    A[1,4] = tilstand
    A[1,5] = kjonn
    if (alder<=72) {
      A[1,6] = 0
      A[1,7] = 0
    }
    if (alder>72 & alder<=74) {
      A[1,6] = 1
      A[1,7] = 0
```

```
  if(!missing(P2)) { P = P2 }
}
if (alder>74) {
  A[1,6] = 0
  A[1,7] = 1
  if(!missing(P3)) { P = P3 }
}
x = rbind(2,3,4)
raten = sum(P[tilstand,])
sann = rbind((P[1,2]/raten),(P[1,3]/raten),(P[1,4]/raten))
temptid = rexp(1,raten)               # Tid brukt i tilstand 1
tid = temptid
alder = alder+temptid
# Hopper til ny tilstand, enten 1,2,3,4
tilstand = sample(x,1,replace=FALSE,sann)
# Andre kolonne
A[2] = i
A[2,2] = alder
A[2,3] = tid
A[2,4] = tilstand
A[2,5] = kjonn
if (alder<=72) {
  A[2,6] = 0
  A[2,7] = 0
}
if (alder>72 & alder<=74) {
  A[2,6] = 1
  A[2,7] = 0
  if(!missing(P2)) { P = P2 }
}
if (alder>74) {
  A[2,6] = 0
  A[2,7] = 1
  if(!missing(P3)) { P = P3 }
}
j = j+2

if(tilstand==2){
  raten = sum(P[tilstand,])
  tilstand = sample(rbind(3,4),1,replace=FALSE,
                    rbind((P[2,3]/raten),(P[2,4]/raten)))
  temptid = rexp(1,raten)
  tid = tid+temptid
  alder = alder+temptid
  B = matrix(data=0, nrow=1, ncol=7, dimnames = list(j,
                    c("ptnum", "alder", "tid", "tilstand",
                    "kjonn", "aldergr2", "aldergr3")))
```

```
  B[1] = i
  B[1,2] = alder
  B[1,3] = tid
  B[1,4] = tilstand
  B[1,5] = kjonn
  if (alder<=72) {
    B[1,6] = 0
    B[1,7] = 0
  }
  if (alder>72 & alder<=74) {
    B[1,6] = 1
    B[1,7] = 0
    if(!missing(P2)) { P = P2 }
  }
  if (alder>74) {
    B[1,6] = 0
    B[1,7] = 1
    if(!missing(P3)) { P = P3 }
  }
  j = j+1
  A = rbind(A,B)
}

if(tilstand==3){
  raten = sum(P[tilstand,])
  tilstand = 4
  temptid = rexp(1,raten)
  tid = tid+temptid
  alder = alder+temptid
  B = matrix(data=0, nrow=1, ncol=7, dimnames = list(j,
                    c("ptnum", "alder", "tid", "tilstand",
                    "kjonn", "aldergr2", "aldergr3")))
  B[1] = i
  B[1,2] = alder
  B[1,3] = tid
  B[1,4] = tilstand
  B[1,5] = kjonn
  if (alder<=72) {
    B[1,6] = 0
    B[1,7] = 0
  }
  if (alder>72 & alder<=74) {
    B[1,6] = 1
    B[1,7] = 0
  }
  if (alder>74) {
    B[1,6] = 0
```

```
        B[1,7] = 1
      }
      j = j+1
      A = rbind(A,B)
    }
    return(A)
  }
  j = 1
  i = 1
  for(i in 1:ant){
    tilstand = 1
    alder = 70
    tid = 0
    kjonn = sample(0:1,1)            # trekker tilfeldig kjønn
    if (missing(V)) { V=Q }

    ## ----- kjonn=0 ----------
    if(kjonn==0 ){
      if(missing(Q2) & missing(Q3)) { A = xFunksjon(Q,i,j) }
      if(!missing(Q2) & !missing(Q3))
        A = xFunksjon(Q,i,j,Q2,Q3)
      j = j+dim(A)[1] # dim(A) gir ant rekker og ant kolonner
    }

    ## ----- kjonn=1 ----------
    if(kjonn==1){
      if(missing(V2) & missing(V3)) { A = xFunksjon(V,i,j) }
      if(!missing(V2) & !missing(V3))
        A = xFunksjon(V,i,j,V2,V3)
      j = j+dim(A)[1]
    }

    if(i==1) X = A
    else X = rbind(X,A)
    i = i+1
  }
  return(X)
}

zMatrise = function(ant,X){
# Tar inn antall individer og en matrise som er blitt generert
# i xMatrise(). Funksjonen ordner da matrisen slik jeg vil
# at den skal se ut. Returnerer en matrise som illustrerer
# en tenkt studie.

i = 1
t = 1
```

```
for(j in 1:ant){
  A = matrix(data=0, nrow=2, ncol=7, dimnames = list(c(t,(t+1)),
                     c("ptnum", "alder", "tid", "tilstand",
                     "kjonn", "aldergr2", "aldergr3")))
  t = t+2
  A[1,] = X[i,]
  A[2,1] = j
  obstid = runif(1,min=0.9,max=1.1)
  A[2,2] = 70+obstid
  A[2,3] = 0+obstid
  A[2,5] = A[1,5]
  A[2,6] = A[1,6]
  A[2,7] = A[1,7]
  i = i+1

  # Hvis alder < enn neste rekke i X vil tilstanden være den samme
  if(A[2,2]<X[i,2]) { A[2,4] = A[1,4] }

  # Og hvis alder >= vil tilstanden endres til den i X
  if(A[2,2]>=X[i,2]) {
    A[2,4] = X[i,4]
    if(A[2,4]==4){
      # hvis død, sett inn det eksakte dødstidspunktet
      A[2,2] = X[i,2]
      A[2,3] = X[i,3]
    }
    i = i+1

    # For å ikke gå utenfor matrisen
    if(i<=dim(X)[1] & A[2,4]!=4){
      if(A[2,2]>=X[i,2]){
        A[2,4] = X[i,4]
        if(A[2,4]==4){
          # hvis død, sett inn det eksakte dødstidspunktet
          A[2,2] = X[i,2]
          A[2,3] = X[i,3]
        }
        i = i+1

        # Igjen, for å ikke gå utenfor matrisen
        if(i<=dim(X)[1] & A[2,4]!=4){
          if(A[2,2]>=X[i,2]){
            A[2,4] = X[i,4]
            if(A[2,4]==4){
              # hvis død, sett inn det eksakte dødstidspunktet
              A[2,2] = X[i,2]
```

```
            A[2,3] = X[i,3]
          }
          i = i+1
        }
      }
    }
  }
}
k = 2

while (A[k,4]<4 & A[k,2]<76){
  B = matrix(data=0, nrow=1, ncol=7, dimnames = list(t,c("ptnum",
                    "alder", "tid", "tilstand", "kjonn",
                    "aldergr2", "aldergr3")))
  t = t+1
  B[1,1] = j
  obstid = obstid+runif(1,min=0.9,max=1.1)
  B[1,2] = 70+obstid
  B[1,3] = 0+obstid
  B[1,5] = A[1,5]

  # Hvis ny alder < alder i neste rekke i X
  if(B[1,2]<X[i,2]) { B[1,4] = A[k,4] }

  # Hvis ny alder >= alder i neste rekke i X
  if(B[1,2]>=X[i,2]){
    B[1,4] = X[i,4]
    i = i+1
    # Sjekke om neste rekke også er under B[1,2] og at den finnes
    if(i<=dim(X)[1] & B[1,4]!=4){
      if(B[1,2]>=X[i,2]){
        B[1,4] = X[i,4]
        i = i+1
        # Igjen, sjekke om neste rekke også er under B[1,2]
        if(i<=dim(X)[1] & B[1,4]!=4){
          if(B[1,2]>=X[i,2]){
            B[1,4] = X[i,4]
            i = i+1
          }
        }
      }
    }
    if(B[1,4]==4){
      # hvis død, sett inn det eksakte dødstidspunktet
      B[1,2] = X[(i-1),2]
      B[1,3] = X[(i-1),3]
    }
```

```
    }

    # Gruppeinndeling
    if (B[1,2]<=72) {
      B[1,6] = 0
      B[1,7] = 0
    }
    if (B[1,2]>72 & B[1,2]<=74) {
      B[1,6] = 1
      B[1,7] = 0
    }
    if (B[1,2]>74) {
      B[1,6] = 0
      B[1,7] = 1
    }
    k = k+1
    A = rbind(A,B)
  }

  if(j<ant) {
    while (X[i,1]==j)
      i = i+1
  }
  else
    i = i+1

  if(j==1) { Z = A }
  else { Z = rbind(Z,A) }
  j = j+1
}
return(Z)
}

feilMatrise = function(Z,E) {
# Tar inn matrisen som ble generert i zMatrise-funksjonen og en
# feilklassifiseringsmatrise. Returnerer en matrise som
# inneholder feile observasjoner.

  t = dim(Z)[1]
  for(i in 1:t){
    if(Z[i,4]==1){
      j = sample(rbind(1,2),1,prob=rbind(E[1,1],E[1,2]))
      Z[i,4] = j
      next
    }
    if(Z[i,4]==2){
      j = sample(rbind(1,2,3),1,prob=rbind(E[2,1],E[2,2],E[2,3]))
```

```
      Z[i,4] = j
      next
    }
    if(Z[i,4]==3){
      j = sample(rbind(2,3),1,prob=rbind(E[3,2],E[3,3]))
      Z[i,4] = j
      next
    }
    if(Z[i,4]==4){ next }
  }
  return(Z)
}
```

A.2 Simuleringer med en intensitetsmatrise

```
library(msm)
library(xtable)

## ---- Simulerer datasett med bare en Q-matrise ------
# Velger antall simuleringer, enten 100 eller 10 000
antSim = 100
# Velger antall individer, enten 500, 200 eller 20
antInd = 200
# Intensitetsmatrisen som brukes for å generere datasettet
q=rbind(c(0,0.3,0.05,0.1),c(0,0,0.2,0.1),c(0,0,0,0.3),c(0,0,0,0))
# qmatrisen som skal sendes inn i msm()-funksjonen
qmatrise = rbind(c(0,0.3,0.1,0.1),c(0,0,0.3,0.2),
                 c(0,0,0,0.4),c(0,0,0,0))
# Lager en matrise for å lagre intensitetene
Qest = matrix(data=0,nrow=antSim,ncol=6)

set.seed(123)

for(s in 1:antSim){
  x = xMatrise(antInd,q)
  z = zMatrise(antInd,x)
  datafil = data.frame(z)
  eks.msm = try(msm(tilstand ~ tid, subject=ptnum,
            data=datafil, qmatrix=qmatrise, death=4),TRUE)
  if(is(eks.msm,"try-error")) {next}
  # Lagrer estimatene fra eks.msm
  Qest[s,1] = eks.msm[[2]]$baseline[1,2]
  Qest[s,2] = eks.msm[[2]]$baseline[1,3]
  Qest[s,3] = eks.msm[[2]]$baseline[1,4]
  Qest[s,4] = eks.msm[[2]]$baseline[2,3]
  Qest[s,5] = eks.msm[[2]]$baseline[2,4]
```

```
  Qest[s,6] = eks.msm[[2]]$baseline[3,4]
  s=s+1
}

# Lagrer en indikator for sjekke om der er noen 0-er (fra
# evt error) og fjerner dem fra matrisen
ind = as.logical(Qest[,1]!=0)
Qest = Qest[ind,]
length(Qest[,1])

# Regner ut MSE
Qmse = matrix(data=0,nrow=1,ncol=6)
Qmse[,1] = mean((q[1,2]-Qest[,1])^2)
Qmse[,2] = mean((q[1,3]-Qest[,2])^2)
Qmse[,3] = mean((q[1,4]-Qest[,3])^2)
Qmse[,4] = mean((q[2,3]-Qest[,6])^2)
Qmse[,5] = mean((q[2,4]-Qest[,5])^2)
Qmse[,6] = mean((q[3,4]-Qest[,6])^2)

# Lagrer estimatene for 500 individer (kjøretid = ca 20 min)
Qmse1 = Qmse
Qsnitt1 = cbind(mean(Qest[,1]),mean(Qest[,2]),mean(Qest[,3]),
                mean(Qest[,4]),mean(Qest[,5]),mean(Qest[,6]))
# Lagrer minimums- og maksimumsverdiene
min1 = cbind(min(Qest[,1]),min(Qest[,2]),min(Qest[,3]),
             min(Qest[,4]),min(Qest[,5]),min(Qest[,6]))
max1 = cbind(max(Qest[,1]),max(Qest[,2]),max(Qest[,3]),
             max(Qest[,4]),max(Qest[,5]),max(Qest[,6]))

# Lagrer estimatene fra 200 individer (kjøretid = ca. 10 min)
Qmse2 = Qmse
Qsnitt2 = cbind(mean(Qest[,1]),mean(Qest[,2]),mean(Qest[,3]),
                mean(Qest[,4]),mean(Qest[,5]),mean(Qest[,6]))
# Lagrer minimums- og maksimumsverdiene
min2 = cbind(min(Qest[,1]),min(Qest[,2]),min(Qest[,3]),
             min(Qest[,4]),min(Qest[,5]),min(Qest[,6]))
max2 = cbind(max(Qest[,1]),max(Qest[,2]),max(Qest[,3]),
             max(Qest[,4]),max(Qest[,5]),max(Qest[,6]))

# Lagrer estimatene fra 20 individer (kjøretid = 2 min, 3 warnings)
Qmse3 = Qmse
Qsnitt3 = cbind(mean(Qest[,1]),mean(Qest[,2]),mean(Qest[,3]),
                mean(Qest[,4]),mean(Qest[,5]),mean(Qest[,6]))

# Lager histogrammer (her: 20 individer)
pdf('histind20linjer.pdf')
par(mfrow=c(3,2))
```

```
x=c("est12","est13", "est14", "est23", "est24", "est34")
for(j in 1:6){
  hist(Qest[,j],main = paste("Histogram of" , x[j]))
  if(antInd==20){
    # Lager vertikale linjer for min og max
    abline(v=min1[,j],lty=2);abline(v=max1[,j],lty=2)
    abline(v=min2[,j],lty=3);abline(v=max2[,j],lty=3)
  }
  j=j+1
}
dev.off()

# Lager tabell
xtable(cbind(t(Qsnitt1),t(Qsnitt2),t(Qsnitt3)),digits=5)
xtable(cbind(t(Qmse1),t(Qmse2),t(Qmse3)),digits=5)

############################################################

### -------------- 10 000 simuleringer --------------------
# Kjører nå det ovenfor med antSim=10000 og følgende set.seed:
set.seed(1) # For 500 ind.
set.seed(6) # For 200 ind.
set.seed(8) # For 20 ind.

# Lagrer resultatene med hhv. 500, 200 og 20 individer
QmseSim1 = Qmse
QsnittSim1 = cbind(mean(Qest[,1]),mean(Qest[,2]),mean(Qest[,3]),
                   mean(Qest[,4]),mean(Qest[,5]),mean(Qest[,6]))
QmseSim2 = Qmse
QsnittSim2 = cbind(mean(Qest[,1]),mean(Qest[,2]),mean(Qest[,3]),
                   mean(Qest[,4]),mean(Qest[,5]),mean(Qest[,6]))
QmseSim3 = Qmse
QsnittSim3 = cbind(mean(Qest[,1]),mean(Qest[,2]),mean(Qest[,3]),
                   mean(Qest[,4]),mean(Qest[,5]),mean(Qest[,6]))
# Lager tabell
xtable(cbind(t(QsnittSim1),t(QmseSim1),t(QsnittSim2),
             t(QmseSim2),t(QsnittSim3),t(QmseSim3)),digits=5)

# Lager en funksjon som kan lage normallinjen
Normalkurve = function(estimat,høyde){
  xv = seq(min(estimat), max(estimat), 0.001)
  yv = dnorm(xv, mean=mean(estimat), sd=sd(estimat))*høyde
  # bredden på søylen: (0.35-0.3)/5=0.01 ---> 10000*0.01=100
  lines(xv,yv)
}

# Lager histogrammer (her: 20 individer)
```

```
pdf('histsimnormal4.pdf')
par(mfrow=c(3,2))
hist(Qest[,1],main = paste("Histogram of" , x[1]))
Normalkurve(Qest[,1],498.2)
hist(Qest[,2],main = paste("Histogram of" , x[2]))
Normalkurve(Qest[,2],498.2)
hist(Qest[,3],main = paste("Histogram of" , x[3]))
Normalkurve(Qest[,3],498.2)
hist(Qest[,4],main = paste("Histogram of" , x[4]))
Normalkurve(Qest[,4],996.4)
hist(Qest[,5],main = paste("Histogram of" , x[5]))
Normalkurve(Qest[,5],996.4)
hist(Qest[,6],main = paste("Histogram of" , x[6]))
Normalkurve(Qest[,6],9964)
dev.off()
```

A.3 Kovariater

```
### ------------ 2 intensitetsmatriser -----------------
# Kjører nå med antInd=500 og antSim=100, og q-matrisen
# gjelder nå for kvinner, mens for menn blir den:
v = rbind(c(0,0.5,0.05,0.1),c(0,0,0.4,0.1),c(0,0,0,0.3),c(0,0,0,0))
# Lager en tilsvarende matrise for å lagre intensitetene for menn
Vest = matrix(data=0,nrow=antSim,ncol=6)

# Simulerer her datasettet ut fra 2 matriser
set.seed(7)
for(s in 1:antSim){
  x <- xMatrise(antInd,q,v)
  z <- zMatrise(antInd,x)
  datafil <- data.frame(z)
  eks.msm <- try(msm(tilstand ~ tid, subject=ptnum, data=datafil,
              qmatrix=qmatrise, death=4, covariates=~kjonn),TRUE)
#  eks.msm <- try(msm(tilstand ~ tid, subject=ptnum, data=datafil,
#                                  qmatrix=qmatrise, death=4),TRUE)
  if(is(eks.msm,"try-error")) {next}
  # Lagrer de estimerte intensitetene
  qtmp <- qmatrix.msm(eks.msm, covariates=list(kjonn=0))
  Qest[s,1]= qtmp[[1]][1,2]
  Qest[s,2]= qtmp[[1]][1,3]
  Qest[s,3]= qtmp[[1]][1,4]
  Qest[s,4]= qtmp[[1]][2,3]
  Qest[s,5]= qtmp[[1]][2,4]
  Qest[s,6]= qtmp[[1]][3,4]
  vtmp <- qmatrix.msm(eks.msm, covariates=list(kjonn=1))
```

```
  Vest[s,1]= vtmp[[1]][1,2]
  Vest[s,2]= vtmp[[1]][1,3]
  Vest[s,3]= vtmp[[1]][1,4]
  Vest[s,4]= vtmp[[1]][2,3]
  Vest[s,5]= vtmp[[1]][2,4]
  Vest[s,6]= vtmp[[1]][3,4]
  s=s+1
}

ind = as.logical(Qest[,1]!=0)
Qest = Qest[ind,]
Vest = Vest[ind,]
length(Qest[,1])

# Regner ut MSE og gjennomsnitt
Qmse = matrix(data=0,nrow=1,ncol=6)
Qmse[,1] = mean((q[1,2]-Qest[,1])^2)
Qmse[,2] = mean((q[1,3]-Qest[,2])^2)
Qmse[,3] = mean((q[1,4]-Qest[,3])^2)
Qmse[,4] = mean((q[2,3]-Qest[,6])^2)
Qmse[,5] = mean((q[2,4]-Qest[,5])^2)
Qmse[,6] = mean((q[3,4]-Qest[,6])^2)
Vmse = matrix(data=0,nrow=1,ncol=6)
Vmse[,1] = mean((v[1,2]-Vest[,1])^2)
Vmse[,2] = mean((v[1,3]-Vest[,2])^2)
Vmse[,3] = mean((v[1,4]-Vest[,3])^2)
Vmse[,4] = mean((v[2,3]-Vest[,6])^2)
Vmse[,5] = mean((v[2,4]-Vest[,5])^2)
Vmse[,6] = mean((v[3,4]-Vest[,6])^2)
Qsnitt = cbind(mean(Qest[,1]),mean(Qest[,2]),mean(Qest[,3]),
               mean(Qest[,4]),mean(Qest[,5]),mean(Qest[,6]))
Vsnitt = cbind(mean(Qest[,1]),mean(Qest[,2]),mean(Qest[,3]),
               mean(Qest[,4]),mean(Qest[,5]),mean(Qest[,6]))
xtable(cbind(t(Qsnitt),t(Qmse),t(Vsnitt),t(Vmse)),digits=5)

# Beta koeffisientene
beta012 = log(Qest[,1])
beta023 = log(Qest[,4])
snittbeta0 = cbind(mean(beta012),mean(beta023))
msebeta0 = cbind(mean((-1.20397-beta012)^2),
                 mean((-1.60944-beta023)^2))
beta12 = log(Qest[,1]/Qest[,1])
beta23 = log(Vest[,4]/Qest[,4])
snittbeta = cbind(mean(beta12),mean(beta23))
msebeta = cbind(mean((0.51083-beta12)^2),mean((0.69315-beta23)^2))
xtable(cbind(t(snittbeta),t(msebeta)),digits=5)
xtable(cbind(t(snittbeta0),t(msebeta0)),digits=5)
```

```
## ------------- 6 intensitetsmatriser -------------
# I tillegg til q og v, lager jeg her fire matriser til, hhv.
# for kvinner og menn
q2 = rbind(c(0,0.42,0.05,0.1),c(0,0,0.3,0.1),c(0,0,0,0.3),c(0,0,0,0))
q3 = rbind(c(0,0.54,0.05,0.1),c(0,0,0.4,0.1),c(0,0,0,0.3),c(0,0,0,0))
v2 = rbind(c(0,0.7,0.05,0.1),c(0,0,0.6,0.1),c(0,0,0,0.3),c(0,0,0,0))
v3 = rbind(c(0,0.9,0.05,0.1),c(0,0,0.8,0.1),c(0,0,0,0.3),c(0,0,0,0))

qmatrise = rbind(c(0,0.3,0.1,0.1),c(0,0,0.3,0.2),
                 c(0,0,0,0.4),c(0,0,0,0))                 # samme
qmatrise = rbind(c(0,0.35,0.25,0.3),c(0,0,0.45,0.3),
                 c(0,0,0,0.5),c(0,0,0,0))                   # nr2

set.seed(23)
x = xMatrise(antInd,q,q2,q3,v,v2,v3)
z = zMatrise(antInd,x)
datafil = data.frame(z)
eks.msm = msm(tilstand ~ tid, subject = ptnum, data = datafil,
       qmatrix=qmatrise, death=4, covariates=~kjonn+gruppe2+gruppe3)

# Spesifiserer control i msm()
# Først for defaulten Nelder-Mead:
eks.msm = msm(tilstand ~ tid, subject = ptnum, data = datafil,
        qmatrix=qmatrise, death=4, covariates=~kjonn+gruppe2+gruppe3,
        control = list(trace=1,REPORT=1))
# Så for BFGS-metoden:
eks.msm = msm(tilstand ~ tid, subject = ptnum, data = datafil,
        qmatrix=qmatrise, death=4, covariates=~kjonn+gruppe2+gruppe3,
        method = "BFGS", control = list(trace=1,REPORT=1))
```

A.4 Feilklassifisering

```
### ----------- Feilklassifisering --------------------

antSim = 100
antInd = 500
# Intensitetsmatrisene som skal brukes for å generere datasettet
q = rbind(c(0,0.3,0.05,0.1),c(0,0,0.2,0.1),
          c(0,0,0,0.3),c(0,0,0,0))                          # samme
c = rbind(c(0.95,0.05,0,0),c(0.05,0.85,0.1,0),
           c(0,0.1,0.9,0),c(0,0,0,1))
# qmatrisen og ematrisen som skal sendes inn i msm()-funksjonen
qmatrise = rbind(c(0,0.3,0.1,0.1),c(0,0,0.3,0.2),
```

```
                  c(0,0,0,0.4),c(0,0,0,0))                  # samme
ematrise = rbind(c(0,0.1,0,0),c(0.1,0,0.1,0),
                  c(0,0.1,0,0),c(0,0,0,0))
Qest = matrix(data=0,nrow=antSim,ncol=6)
Cest = matrix(data=0,nrow=antSim,ncol=4)

set.seed(100)
for(s in 1:antSim){
  x = xMatrise(antInd,q)
  z = zMatrise(antInd,x)
  feilz = feilMatrise(z,c)
  datafil = data.frame(feilz)
  eks.msm = try(msm(tilstand ~ tid, subject=ptnum, data=datafil,
                    qmatrix=qmatrise, ematrix=ematrise, death=4,
                    method="BFGS"),TRUE)
  if(is(eks.msm,"try-error")) {next}
  # henter ut intensitetsestimatene
  Qest[s,1] = eks.msm[[2]]$baseline[1,2]
  Qest[s,2] = eks.msm[[2]]$baseline[1,3]
  Qest[s,3] = eks.msm[[2]]$baseline[1,4]
  Qest[s,4] = eks.msm[[2]]$baseline[2,3]
  Qest[s,5] = eks.msm[[2]]$baseline[2,4]
  Qest[s,6] = eks.msm[[2]]$baseline[3,4]
  # henter ut feilklassifiseringsestimatene
  Cest[s,1] = eks.msm[[26]]$baseline[1,2]
  Cest[s,2] = eks.msm[[26]]$baseline[2,1]
  Cest[s,3] = eks.msm[[26]]$baseline[2,3]
  Cest[s,4] = eks.msm[[26]]$baseline[3,2]
  s = s+1
}

ind = as.logical(Qest[,1]!=0)
Qest = Qest[ind,]
Vest = Vest[ind,]
length(Qest[,1])

# Regner ut MSE
Qmse = matrix(data=0,nrow=1,ncol=6)
Qmse[,1] = mean((q[1,2]-Qest[,1])^2)
Qmse[,2] = mean((q[1,3]-Qest[,2])^2)
Qmse[,3] = mean((q[1,4]-Qest[,3])^2)
Qmse[,4] = mean((q[2,3]-Qest[,6])^2)
Qmse[,5] = mean((q[2,4]-Qest[,5])^2)
Qmse[,6] = mean((q[3,4]-Qest[,6])^2)
Cmse = matrix(data=0,nrow=1,ncol=4)
Cmse[,1] = mean((c[1,2]-Cest[,1])^2)
Cmse[,2] = mean((c[1,3]-Cest[,2])^2)
```

```
Cmse[,3] = mean((c[2,3]-Cest[,3])^2)
Cmse[,4] = mean((c[3,4]-Cest[,4])^2)
Qsnitt = cbind(mean(Qest[,1]),mean(Qest[,2]),mean(Qest[,3]),
               mean(Qest[,4]),mean(Qest[,5]),mean(Qest[,6]))
Csnitt = cbind(mean(Cest[,1]),mean(Cest[,2]),mean(Cest[,3]),
                  mean(Cest[,4]))
xtable(cbind(t(Qsnitt), t(Qmse)),digits=5)
xtable(cbind(t(Csnitt), t(Cmse)),digits=5)

# Brukt på den første simuleringen
statetable.msm(tilstand, ptnum, datafil)
```

A.5 Utprøving av feilklassifiseringsmatrisen

```
### --------  Endrer på C-matrisen (tilfører flere feil) -----------

# Endringer
c = rbind(c(0.8,0.2,0,0),c(0.1,0.8,0.1,0),c(0,0.2,0.8,0),c(0,0,0,1))
c = rbind(c(0.8,0.2,0,0),c(0.2,0.6,0.2,0),c(0,0.2,0.8,0),c(0,0,0,1))
c = rbind(c(0.7,0.3,0,0),c(0.2,0.6,0.2,0),c(0,0.3,0.7,0),c(0,0,0,1))
c = rbind(c(0.6,0.4,0,0),c(0.2,0.6,0.2,0),c(0,0.4,0.6,0),c(0,0,0,1))
c = rbind(c(0.5,0.5,0,0),c(0.2,0.6,0.2,0),c(0,0.5,0.5,0),c(0,0,0,1))
c = rbind(c(0.5,0.5,0,0),c(0.3,0.4,0.3,0),c(0,0.5,0.5,0),c(0,0,0,1))
c = rbind(c(0.5,0.5,0,0),c(0.25,0.5,0.25,0),
          c(0,0.5,0.5,0),c(0,0,0,1))
c = rbind(c(0.4,0.6,0,0),c(0.2,0.6,0.2,0),c(0,0.6,0.4,0),c(0,0,0,1))
c = rbind(c(0.6,0.4,0,0),c(0.2,0.6,0.2,0),c(0,0.6,0.4,0),c(0,0,0,1))
c = rbind(c(0.4,0.6,0,0),c(0.2,0.6,0.2,0),c(0,0.4,0.6,0),c(0,0,0,1))
c = rbind(c(0.7,0.3,0,0),c(0.2,0.6,0.2,0),c(0,0.7,0.3,0),c(0,0,0,1))

feilz = feilMatrise(z,c)
datafil = data.frame(feilz)
statetable.msm(tilstand,ptnum,datafil)
eks.msm = msm(tilstand ~ tid, subject=ptnum, data=datasett,
          qmatrix=qmatrise, ematrix=ematrise, death=4, method="BFGS")

# Henter ut verdiene
est = matrix(data=0,nrow=1,ncol=10)
est[1] = eks.msm[[2]]$baseline[1,2]
est[2] = eks.msm[[2]]$baseline[1,3]
est[3] = es.msm[[2]]$baseline[1,4]
est[4] = eks.msm[[2]]$baseline[2,3]
est[5] = eks.msm[[2]]$baseline[2,4]
est[6] = eks.msm[[2]]$baseline[3,4]
est[7] = eks.msm[[26]]$baseline[1,2]
est[8] = eks.msm[[26]]$baseline[2,1]
```

```
est[9] = eks.msm[[26]]$baseline[2,3]
est[10] = eks.msm[[26]]$baseline[3,2]
xtable(t(est),digits=5)
```

Bibliografi

[1] Alexandre Bureau, Stephen Shiboski, and James P. Hughes. Applications of continuous time hidden markov models to the study of misclassified disease outcomes. *Statistics in Medicine*, 22(3):441–462, 2003.

[2] Olivier Cappé, Eric Moulines, and Tobias Rydén. *Inference in Hidden Markov Models*. Springer Science+Business Media, 1st edition, 2005.

[3] George Casella and Roger L. Berger. *Statistical Inference*. Duxbury, Thompson Learning Inc, 2nd edition, 2002.

[4] John E. Dennis and Robert B. Schnabel. *Numerical methods for unconstrained optimization and nonlinear equations*. Prentice-Hall, Inc., 1983.

[5] Annette J. Dobson and Adrian G. Barnett. *An Introduction to Generalized Linear Models*. CRC Press, 3rd edition, 2008.

[6] Richard Durbin, Sean R. Eddy, Anders Krogh, and Graeme Mitchison. *Biological sequence analysis: Probabilistic models of proteins and nucleic acids*. United Kingdom: Cambridge University Press, 1998.

[7] Christopher H. Jackson. Package 'msm'. http://cran.r-project.org/web/packages/msm/index.html.

[8] Christopher H. Jackson. Multi-state modelling with R: the msm package. 1.0.1:59, 2011.

[9] Christopher H. Jackson and Linda D. Sharples. Hidden markov models for the onset and progression of bronchiolitis obliterans syndrome in lung transplant recipients. *Statistics in Medicine*, 21(1):113–128, 2002.

[10] Christopher H. Jackson, Linda D. Sharples, Simon G. Thompson, Stephen W. Duffy, and Elisabeth Couto. Multistate markov models for disease progression with classification error. *Journal of the Royal Statistical Society. Series D (The Statistician)*, 52(2):193–209, 2003.

[11] B. H. Juang and L. R. Rabiner. Hidden markov models for speech recognition. *Technometrics*, 33(3):251–272, 1991.

[12] David Kincaid and Ward Cheney. *Numerical Analysis: mathematics of scientific computing*. American Mathematical Society, 3rd edition, 2002.

[13] Stein Atle Lie. *Survival studies of total hip replacements and postoperative mortality.* PhD thesis, Department of Public Health and Primary Health Care, University of Bergen, 2001.

[14] J. A. Nelder and R. Mead. A simplex method for function minimization. *The Computer Journal*, 7(4):308–313, 1965.

[15] Lee Newberg. Error statistics of hidden markov model and hidden boltzmann model results. *BMC Bioinformatics*, 10(1):212, 2009.

[16] Glen A. Satten and Jr Longini, Ira M. Markov chains with measurement error: Estimating the 'true' course of a marker of the progression of human immunodeficiency virus disease. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 45(3):275–309, 1996.

[17] Howard M. Taylor and Samuel Karlin. *An Introduction To Stochastic Modeling.* Academic Press, 3rd edition, 1998.

[18] Andrew C. Titman and Linda D. Sharples. A general goodness-of-fit test for markov and hidden markov models. *Statistics in Medicine*, 27(12):2177–2195, 2008.

[19] Ardo Van Den Hout, Carol Jagger, and Fiona E. Matthews. Estimating life expectancy in health and ill health by using a hidden markov model. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 58(4):449–465, 2009.

[20] Ardo Van Den Hout and Fiona E. Matthews. Estimating stroke-free and total life expectancy in the presence of non-ignorable missing values. *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, 173(2):331–349, 2010.

[21] Walter Zucchini and Iain L. MacDonald. *Hidden Markov models for time series : an introduction using R*, volume Monographs on statistics and applied probability 110. Boca Raton, Fla. : CRC Press, 2009.