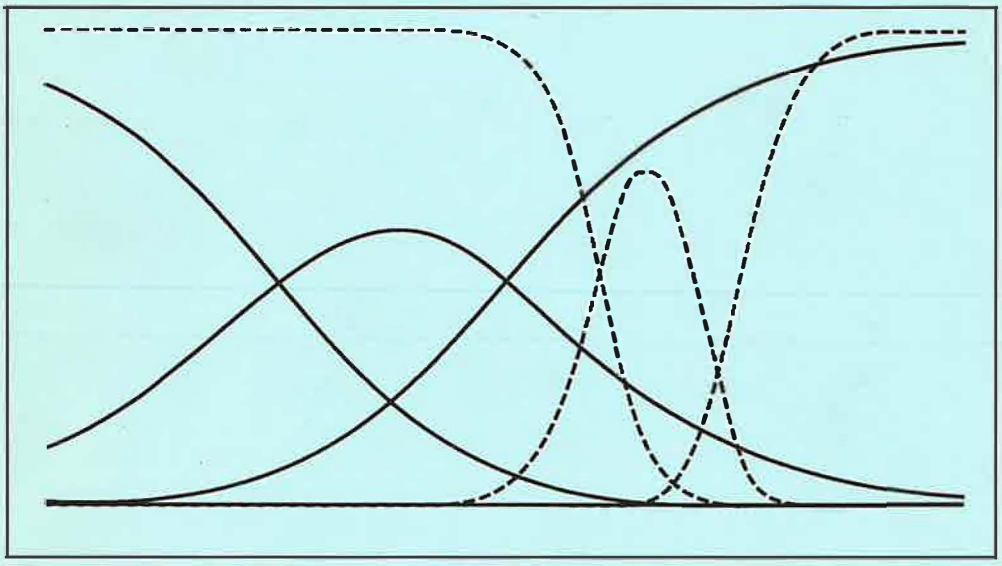
OPD Memorandum 92-3

HET EENPARAMETER LOGISTISCH MODEL (OPLM)

EEN THEORETISCHE INLEIDING EN EEN HANDLEIDING BIJ HET COMPUTERPROGRAMMA



N.D. Verhelst

Cito

HET EENPARAMETER LOGISTISCHE MODEL (OPLM)

Een theoretische inleiding en een handleiding bij het computerprogramma

N.D. Verhelst

Arnhem, juli 1992
eerste druk

Prijs ƒ 35,--

VOORWOORD

Voorliggend memorandum is de neerslag van een cursus itemresponstheorie (IRT) die aan vakmedewerkers van het Cito wordt gegeven. Veel van de cursisten hebben een alpha-opleiding en lopen doorgaans niet erg warm voor de elegantie van wiskundige argumenten. Bij het opstellen van de cursus is gepoogd met dit gegeven consequent rekening te houden. Derhalve zal men in de tekst geen wiskundige bewijzen aantreffen. Belangrijke begrippen, zoals bijvoorbeeld conditionele grootste-aannemelijkheidsschatters, worden geïllustreerd aan de hand van een klein voorbeeld terwijl een exacte definitie achterwege gelaten wordt. Bij het presenteren en hanteren van formules wordt vooral de nadruk gelegd op de interpretatie van de onderdelen van de formule. Voorliggend werk moet dan ook niet beschouwd worden als een leerboek voor de psychometrie; het is een eerste kennismaking met een ingewikkelde statistische theorie voor lezers die 'de essentie willen begrijpen'.

Het cursusmateriaal is zeer specifiek gehouden en gericht op de modellen die op het Cito zo ongeveer de huisstandaard zijn geworden: het Rasch-model en de op het Cito ontwikkelde uitbreiding, het éénparameter logistisch model (OPLM). Naast de theoretische uiteenzettingen, zijn ook drie hoofdstukken opgenomen die beschouwd kunnen worden als de handleiding voor het op het CITO ontwikkelde programmapakket OPLM. Het zijn de hoofdstukken 3, 6 en 8. Deze hoofdstukken kunnen ook als een op zichzelf staande handleiding gebruikt worden.

Mocht bij lezing van de tekst de indruk ontstaan dat IRT op het Cito is uitgevonden, of dat IRT uitsluitend bestaat uit OPLM, dan wil ik hier benadrukken dat deze indruk niet met de werkelijkheid overeenkomt. Zelf heb ik het vak geleerd uit deze twee onvolprezen boeken:

- Fischer, G.H. (1974). Einführung in die Theorie psychologischer Tests. Bern: Huber.

- Lord F.M.& Novick M.R. (1968). Statistical theories of mental test scores. Reading: Addison-Wesley.

Iets minder moeilijke en meer op de praktijk afgestemde inleidingen zijn

- Andrich D. (1988). Rasch models for measurement. Newbury Park: Sage.

- Hambleton, R.K. & Swaminathan, H. (1985). Item response theory. Boston: Kluwer-Nijhoff.

- Lord, F.M. (1980). Applications of item response theory to practical testing problems. Hillsdale, NJ: Erlbaum.

Dit memorandum is niet in de eenzaamheid van een zolderkamertje tot stand gekomen. Het juiste gebruik en een heldere interpretatie van statistische modellen leert men niet alleen uit de boeken, maar ook in gesprekken en discussies met collega's. Mijn dank gaat dan ook uit naar de collega's van de afdeling Onderzoek en Psychometrische Dienstverlening van het Cito met wie ik veel discussies heb gevoerd over een 'klant-vriendelijke' presentatie van een ingewikkelde theorie. Speciale dank ben ik verschuldigd aan W. van Roosmalen en N. Veldhuijzen voor hun taalkundige en didactische adviezen en aan W. Kuijsten voor de tekstverzorging. De onvolkomenheden die de lezer wellicht nog zal ontdekken zijn echter geheel voor mijn rekening.

Norman Verhelst
Arnhem, 24 juli 1992

INHOUDSOPGAVE

1 Latente variabelen en itemresponsfuncties

1.1 Kritiek op de klassieke testtheorie

Indien bij een persoon meerdere malen een toets wordt afgenomen, zal niemand verwachten dat het resultaat elke keer precies hetzelfde zal zijn, ook niet in gevallen waar geheugen- of leereffecten geen enkele rol spelen. Men houdt er dus rekening mee dat de toetsscore (X) niet perfect de kennis of vaardigheid van de geteste persoon weerspiegelt, of met andere woorden dat er een meetfout (E) is gemaakt. In de klassieke testtheorie is men geïnteresseerd in de 'ware' score (T), en men hanteert als basisvergelijking

$$X = T + E \qquad (1)$$

waarbij men eist dat de gemiddelde of verwachte fout gelijk is aan nul. De ware score zou men in principe kunnen achterhalen door de toets een zeer groot aantal keren bij dezelfde persoon af te nemen onder exact dezelfde condities en van al de geobserveerde scores het gemiddelde te bepalen. Zo'n procedure is in de praktijk natuurlijk niet uitvoerbaar. In de klassieke testtheorie heeft men slimme manieren gevonden om met andere, wel uitvoerbare procedures, iets zinnigs te kunnen zeggen over de (gemiddelde) grootte van de meetfout. Op deze procedures wordt hier verder niet ingegaan. We zullen ons beperken tot een meer fundamentele moeilijkheid waar de klassieke testtheorie mee te kampen heeft.

Een toets bestaat doorgaans uit meerdere onderdelen (items) en op elk onderdeel kan men een aantal 'punten' behalen. De toetsscore is dan het totaal aantal behaalde 'punten' op alle onderdelen samen, of met andere woorden de toetsscore ontstaat door het simpelweg optellen van de itemscores. Deze operatie kan men toepassen op een willekeurige verzameling items, waarbij het maximum te behalen punten op elk van die items ook nog eens willekeurig bepaald kan worden. In de klassieke testtheorie verklaart men formule (1) gewoon van toepassing en indien men denkt aan de procedure van een zeer groot aantal toetsafnames, dan ziet men dadelijk in dat (1) ook werkelijk van toepassing is. Dit betekent dat de basisvergelijking van

de klassieke testtheorie altijd van toepassing is en de klassieke testtheorie dus eigenlijk niet het epitheton 'theorie' verdient, omdat ze nooit gefalsificeerd kan worden. Wat betekent dit concreet?

(1) De klassieke testtheorie kan niet aangeven of het zinvol is een gegeven collectie items tot een toets samen te voegen.
(2) Uit de theorie volgt niet hoe men de items dient te scoren.
(3) De theorie rechtvaardigt niet dat het zinvol is de itemscores bij elkaar op te tellen. Men zou bijvoorbeeld ook eerst enkele itemscores met elkaar kunnen vermenigvuldigen vooraleer op te tellen.

In de moderne testtheorie, die vaak aangeduid wordt met de naam itemresponstheorie (IRT), poogt men een theorie op te stellen die op deze kritiek op de klassieke testtheorie een adekwaat antwoord geeft. We komen op deze kwestie terug in hoofdstuk 5.

1.2 Latente variabelen en itemresponsfuncties

In de IRT speelt het begrip 'ware toetsscore' een zeer ondergeschikte rol. Centraal in de IRT staat een abstract begrip dat men zou kunnen aanduiden als 'vaardigheid'. Men gaat ervan uit dat personen verschillen in vaardigheid (de vaardigheid is een variabele) en dat het op één of andere manier mogelijk moet zijn aan elke persoon een getal toe te kennen dat zijn vaardigheid adekwaat weerspiegelt. Bovendien is men ervan overtuigd dat de hoeveelheid vaardigheid van een persoon niet direct waarneembaar is. Vandaar dat men die vaardigheid een niet-observeerbare of latente variabele noemt. Wij zullen die vaardigheid voorstellen met het symbool ϑ. Bovendien zullen we geen begrenzing van de vaardigheid veronderstellen ($-\infty < \vartheta < \infty$). Om toch iets te kunnen zeggen over die vaardigheid veronderstelt men dat de antwoorden op bepaalde items enige indicatie geven over die vaardigheid. Bijvoorbeeld door een uitspraak als: 'een correct antwoord op deze som duidt op een grotere rekenvaardigheid dan een foutief antwoord.' Met zo'n vage uitspraak kunnen we natuurlijk niet veel doen. In de IRT wordt het verband tussen de latente variabele en de itemantwoorden heel expliciet gemaakt. Het centrale begrip hier is de itemresponsfunctie.

Eerst een definitie. Met X_i duiden we de antwoordvariabele aan, en voorlopig gaan we ervan uit dat deze antwoordvariabele slechts twee waarden kan aannemen volgens de regel

$$X_i = \begin{cases} 1 \text{ indien het antwoord op item i correct is,} \\ 0 \text{ indien het antwoord op item i fout is.} \end{cases}$$

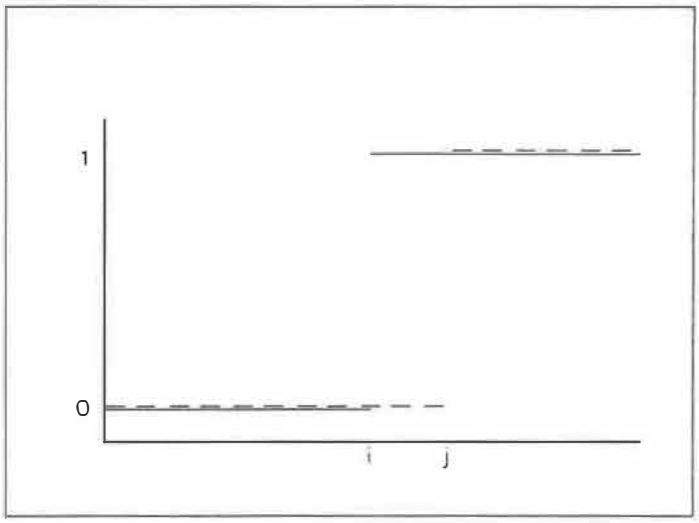
Centraal in de IRT is de aanname dat het antwoord op een item nooit volledig vastligt, hoe groot of hoe klein de vaardigheid van de persoon die antwoordt ook is. Daarom werkt men in de IRT met kansen. De itemrespons-functie drukt uit hoe groot de kans is dat een item juist wordt beantwoordt, als functie van de vaardigheid ϑ. De itemresponsfunctie duiden we aan met het symbool $f_i(\vartheta)$. Deze functie geeft aan wat de kans is op een juist antwoord op item i voor iemand die een vaardigheid ϑ heeft. Dus:

$$f_i(\vartheta) = \text{Prob}(X_i=1|\vartheta) \qquad (2)$$

De itemresponsfunctie is dus een conditionele kans en vertelt ons iets over het gedrag van de persoon, als we zijn vaardigheid kennen. Bemerk dat de theorie eigenlijk aan de verkeerde kant begint. Uiteindelijk zijn wij geïnteresseerd in uitspraken over de vaardigheid, als we het gedrag kennen.

Formule (2) is nog geen theorie; het is eigenlijk niets meer dan een conventie over de notatie: we schrijven kortheidshalve het linkerlid op, als we het rechterlid bedoelen. Om een echte theorie te maken zullen we de functie moeten specificeren, dat wil zeggen we moeten het verloop ervan beschrijven en er de eigenschappen van vastleggen. Omdat we later mathematische manipulaties met die functie zullen moeten uitvoeren, zullen we eisen dat ze niet te 'gek' is en dat ze 'geloofwaardig' is. We beginnen met een niet-geloofwaardige functie, die als volgt geconstrueerd wordt. Voor een item i veronderstelt men dat er een bepaalde hoeveelheid vaardigheid nodig is om een correct antwoord te produceren. Iemand die over minder vaardigheid beschikt zal, nooit een correct antwoord geven (de kans op een correct antwoord is nul), terwijl iemand met meer vaardigheid het item altijd juist beantwoordt (dat wil zeggen met kans 1). De grafiek van

de itemresponsfunctie is weergegeven in Figuur 1. In dezelfde figuur is ook de plaats aangegeven voor een moeilijker item j. (Item j is moeilijker dan item i, omdat de minimale vaardigheid vereist voor een correct antwoord op item j groter is dan voor item i; de grafiek van $f_j(\vartheta)$ springt van 0 naar 1 op de plaats j in de Figuur.)



Figuur 1

Itemresponsfunctie
in een deterministisch model

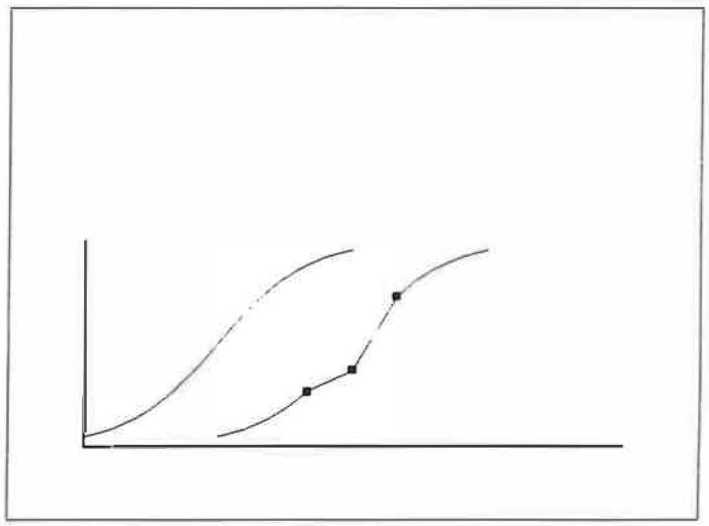
Deze theorie ziet er misschien aantrekkelijk uit, want ze impliceert het principe: 'wie een moeilijk item (j) juist beantwoordt, geeft ook een juist antwoord op een gemakkelijker item (i)'. Ze is echter niet geloofwaardig, omdat het in de praktijk bijna nooit voorkomt dat er in de steekproef niemand is die dit principe schendt. Eén inbreuk op dit principe is voldoende om de theorie te verwerpen. Uit inspectie van Figuur 1 konden we eigenlijk al dit soort moeilijkheden verwachten. Omdat de kans op een juist antwoord altijd precies nul of één is, leggen we de waarde van X_i volledig vast, als we ϑ kennen, en in de praktijk kunnen we daarvoor gestraft worden. Daarom werkt men in de IRT meestal met itemresponsfuncties die nooit exact de waarde 0 of 1 aannemen. Een andere eigenschap die de functie in Figuur 1 onrealistisch maakt is de sprong op een bepaald punt (de functie is discontinu). Wat we dan wel weer als een realistische eigenschap

kunnen beschouwen, is dat de functie in Figuur 1 nooit daalt: de kans op een juist antwoord wordt nooit kleiner als de vaardigheid toeneemt. We gaan deze eigenschap nog iets aanscherpen door te eisen dat de functie overal stijgend moet zijn (dat wil zeggen ze mag niet constant blijven in een bepaald gebied). Samengevat stellen we de volgende eisen aan de item-respons-funcie:

(1) $0 < f_i(\vartheta) < 1$;

(2) de functie is continue ('de grafiek moet getekend kunnen worden zonder de pen op te tillen');

(3) de functie is strikt stijgend.



Figuur 2

Een 'vloeiende' en een 'hoekige' itemresponsfunctie

Figuur 2 toont twee grafieken die aan die drie eisen voldoen. Een eigenschap die die twee grafieken onderscheidt is de 'hoekigheid'. Functies die dit soort hoekigheid vertonen zijn wiskundig meestal niet elegant om mee te werken. Daarom sluiten we hoekige functies uit door een vierde eis:

(4) de functie moet een vloeiend verloop hebben. (Technisch: de functie moet differentieerbaar zijn.)

Hoewel de vier gestelde eisen een groot aantal functies uitsluiten, blijven er nog heel veel functies over die aan alle gestelde eisen voldoen. Door één specifieke functie te kiezen perkt men de theorie verder in tot één speciaal geval. Zo'n speciaal geval noemt men een (IRT-)model. Een specifieke keuze baseert men op een veelheid aan argumenten. Op deze argumenten gaan we hier niet in, tenzij door op te merken dat (mathematische) hanteerbaarheid vaak een belangrijke overweging is. In de volgende paragraaf bespreken we één specifiek geval: het Rasch-model.

1.3 Het Rasch-model

In het Rasch-model is de itemresponsfunctie een logistische functie. De logistische functie (van een argument y) wordt gedefinieerd als

$$f(y) = \frac{e^y}{1+e^y}. \qquad (3)$$

In het Rasch-model is het argument van de logistische functie het verschil $(\vartheta-\sigma_i)$, waarbij σ_i een kengetal is dat item i karakteriseert. Vervangen we nu in het rechterlid van (3) het argument y door dit verschil, dan krijgen we

$$f_i(\vartheta) = \frac{e^{\vartheta-\sigma_i}}{1+e^{\vartheta-\sigma_i}}. \qquad (4)$$

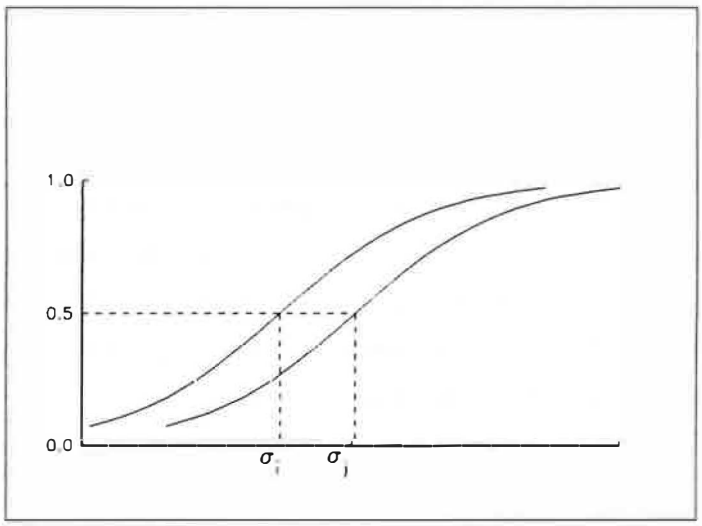
Het zal duidelijk zijn dat door de waarde van σ_i te veranderen een andere functie ontstaat. Omdat we nu nog niets willen zeggen over de preciese waarde van σ_i, definieert (4) in feite een hele familie van functies die allemaal aan de logistische functieregel voldoen. We doen een eenvoudig functieonderzoek van (4). Het is gemakkelijk na te gaan dat de logistische functie f(y) altijd tussen 0 en 1 ligt. Bovendien geldt dat $f(0) = \frac{1}{2}$. Dus geldt dat

$$f_i(\sigma_i) = 1/2 \qquad (5a)$$

Het is bovendien eenvoudig na te gaan dat de volgende twee limieten gelden:

$$\begin{cases} \lim_{\vartheta \to \infty} f_i(\vartheta) = 1, \\ \lim_{\vartheta \to -\infty} f_i(\vartheta) = 0. \end{cases} \qquad (5b)$$

In Figuur 3 staan twee itemresponsfuncties afgebeeld. Twee punten van commentaar op bovenstaand functie onderzoek. (5a) betekent dat, indien de vaardigheid precies gelijk is aan het getal σ_i, de kans op een juist antwoord precies ½ is. Omgekeerd kunnen we σ_i interpreteren als de hoeveelheid vaardigheid die nodig is om een kans te hebben van ½ op een juist antwoord. In Figuur 3 zien we dat meer vaardigheid vereist is om die kans te halen bij item j dan bij item i. Het is dus gerechtvaardigd om te zeggen dat σ_i de moeilijkheid uitdrukt van item i. σ_i wordt daarom vaak de moeilijkheidsparameter van het item genoemd. Omdat er met elk item slechts een parameter gemoeid is, wordt σ_i ook vaak kortweg de itemparameter genoemd.



Figuur 3

Twee itemresponsfuncties

Het tweede commentaar heeft betrekking op (5b). Voor zeer kleine waarden van ϑ is de kans ongeveer nul dat een correct antwoord wordt gegeven. Dit betekent dat het Rasch-model eigenlijk ongeschikt is voor meerkeuzevragen. Iemand die hoegenaamd niets weet over het gevraagde onderwerp heeft bij meerkeuzevragen een substantiële kans op een juist antwoord.

Een inspectie van Figuur 3 laat zien dat de twee curven een identieke vorm hebben; ze zijn alleen verschoven ten opzichte van elkaar. Dit betekent ook dat ze elkaar nooit kruisen. Daaruit volgt dat $f_i(\vartheta) > f_j(\vartheta)$ voor elke waarde van ϑ. In woorden: wat ook de waarde van ϑ is, de kans om item i juist te maken is steeds groter dan de kans on item j juist te maken.

1.4 Lokale stochastische onafhankelijkheid

Formule (4) beschrijft het gedrag van iemand met vaardigheid ϑ op één item. Dit is echter niet voldoende om het Rasch-model te karakteriseren. Er moet ook nog iets gezegd worden over het gedrag, indien meerdere items moeten worden beantwoord. Stel dat we over vier items beschikken die precies even moeilijk zijn, en we leggen die items voor aan twee personen waarvan we weten dat ze dezelfde ϑ-waarde hebben. Na het beantwoorden van de eerste drie items stellen we vast dat de eerste persoon drie juiste antwoorden heeft gegeven en de tweede persoon drie onjuiste. Is het dan niet redelijk te veronderstellen dat de eerste persoon een grotere kans heeft om het vierde item juist te maken dan de tweede persoon? De eerste persoon heeft immers er blijk van gegeven vaardiger te zijn dan de tweede, gezien zijn drie juiste antwoorden. Het antwoord luidt: neen. Immers, als we aannemen dat het Rasch-model geldig is, dan hangt de kans op een juist antwoord alleen af van de vaardigheid en de moeilijkheid van het item, en in de beschreven situatie gaat het om hetzelfde item en dezelfde vaardigheid. Dus moeten die kansen gelijk zijn. Kennis van antwoorden op andere items kan die kans niet veranderen. Deze redenering volgt niet automatisch uit formule (4); ze wordt toegevoegd als een onafhankelijk principe of axioma, namelijk het axioma der lokale stochastische onafhankelijkheid. Dit principe kan op verschillende equivalente manieren in formulevorm worden uitgedrukt. We geven twee belangrijke formules. De antwoordvariabelen X_i en X_j zijn lokaal stochastisch onafhankelijk (van elkaar) indien

$$\text{Prob}(X_i=1 \,|\, \vartheta \text{ en } X_j=1) = \text{Prob}(X_i=1 \,|\, \vartheta) = f_i(\vartheta), \qquad (6a)$$

of

$$\text{Prob}(X_i=1 \text{ en } X_j=1|\vartheta) = \text{Prob}(X_i=1|\vartheta)\,\text{Prob}(X_j=1|\vartheta) = f_i(\vartheta)\,f_j(\vartheta). \quad (6b)$$

(Let wel (6a) en (6b) zijn niet twee verschillende voorwaarden; ze zijn equivalent en betekenen dus precies hetzelfde.) De beperking 'lokaal' wijst erop dat X_i en X_j alleen onafhankelijk zijn bij gelijke ϑ. Daaruit volgt niet dat X_i en X_j onafhankelijk zijn van elkaar. Dus uit lokale stochastische onafhankelijkheid volgt niet dat

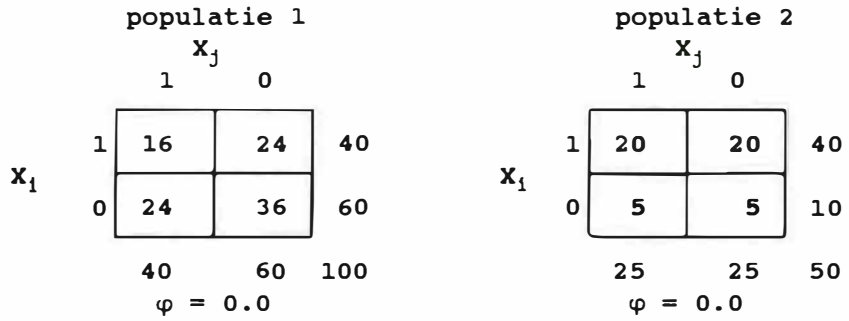
$$\text{Prob}(X_i=1 \text{ en } X_j=1) = \text{Prob}(X_i=1)\,\text{Prob}(X_j=1).$$

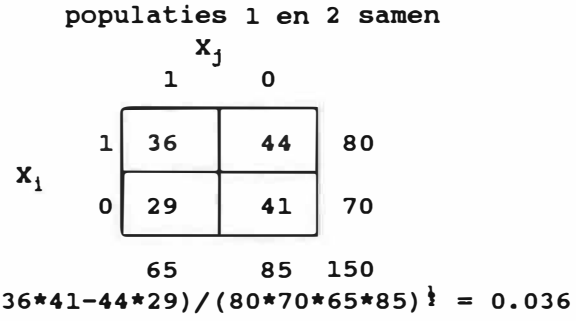
Immers, indien dit waar zou zijn, dan zou de correlatie tussen (de antwoorden op) item i en item j nul bedragen, iets wat in het algemeen niet waar is. Het principe van de lokale stochastische onafhankelijkheid impliceert wel dat de correlatie tussen X_i en X_j nul is in alle populaties waar ϑ constant is. Dit geeft ons meteen een aardige manier om de correlatie tussen items te verklaren: als in een populatie de correlatie tussen item i en j niet nul is, dan komt dat, doordat de vaardigheid in die populatie niet constant is. Door de invloed van de vaardigheid te controleren (dat wil zeggen door de vaardigheid constant te houden) verdwijnt de correlatie. We illustreren dit aan de hand van een voorbeeld. (Zie Figuur 4.)
Het voorbeeld in Figuur 4 laat zien dat de variabelen X_i en X_j niet correleren in populatie 1 noch in populatie 2. Voegen we de twee populaties echter samen, dan wordt de correlatie positief.

Het axioma van de lokale stochastische onafhankelijkheid is zeer belangrijk in de IRT, maar het is erg moeilijk om te controleren of eraan voldaan is. We kunnen namelijk niet te werk gaan op de manier zoals weergegeven in Figuur 4. Dit zou vereisen dat we de totale steekproef zouden kunnen opdelen in groepjes personen die dezelfde ϑ-waarde hebben. Doch ϑ kennen we niet, dus is deze benadering onmogelijk. Voor de toetsconstructeur is het belangrijk dit axioma niet te schenden door items te maken die functioneel afhankelijk zijn van elkaar. Een voorbeeld van twee items die functioneel afhankelijk zijn:

item 1. Hoeveel is 7^2?

item 2. Hoeveel is de derdemachtswortel uit 7^2?





Figuur 4

Een voorbeeld van lokale stochastische onafhankelijkheid

Iemand die niet kan kwadrateren heeft beide items fout, ook al weet hij hoe een derdemachtswortel moet uitgerekend worden. Dus, indien het eerste item fout is beantwoord, is de kans op een juist antwoord op item 2 nul. Of nog anders, een juist antwoord op item 2 impliceert een juist antwoord op item 1. Er kunnen echter veel subtielere vormen van functionele afhankelijkheid bestaan. Het is denkbaar dat door een correct antwoord op een bepaald item informatie vrijkomt die aangewend kan worden voor de beantwoording van een ander item. Dit kan reeds voldoende zijn om de lokale stochastische onafhankelijkheid te schenden. Een goede werkwijze bij het construeren van een toets bestaat erin zich af te vragen of de moeilijkheid van elk item zou veranderen door het uit de context van de andere items los te maken, of door de items in een andere volgorde te presenteren. Indien hierop bevestigend moet worden geantwoord, is het bijna zeker dat aan de eis van lokale stochastische onafhankelijkheid niet voldaan is.

2 Het schatten van de itemparameters in het Rasch-model

2.1 Grootste-aannemelijkheidsschatters: een voorbeeld

Door het Rasch-model als model voor het beantwoorden van de items aan te nemen zijn we natuurlijk nog niet klaar met het werk. Om formule (4) uit te rekenen moeten we een getalswaarde invullen voor ϑ en voor σ_i en die getallen kennen we niet. ϑ en σ_i worden parameters genoemd en men gebruikt de observaties om schattingen te maken van de parameters.

Er zijn verschillende manieren om parameters te schatten. Hier wordt er één besproken, namelijk de grootste-aannemelijkheidsmethode (In het Engels: Maximum Likelihood, afgekort als ML.) We leggen de methode uit aan de hand van een voorbeeld. Een onzuiver muntstuk wordt vijf maal opgegooid, waarbij de uitkomst 'Munt' als een succes beschouwd wordt en de uitkomst 'Kruis' als een mislukking. We definiëren weer toevalsvariabelen X_i als

$$X_i = \begin{cases} 1 \text{ indien 'Munt' bij de i-de beurt,} \\ 0 \text{ indien 'Kruis' bij de i-de beurt } (i=1,\ldots,5). \end{cases}$$

Het model is zeer simpel. Het zegt dat de kans op succes bij opgooien gelijk is aan π, waarbij π een getal is tussen 0 en 1. Wij willen de uitkomsten van ons kleine experimentje gebruiken om π te schatten. Stel dat we de volgende uitkomsten waarnemen: (1 0 1 1 0 Dè. probabiliteit van die uitkomst is

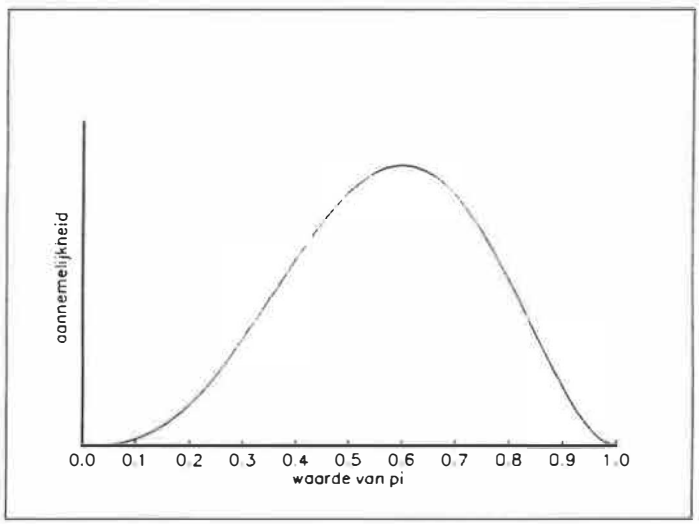
$$\begin{aligned} \text{Prob}(X_1=1, X_2=0, X_3=1, X_4=1, X_5=0; \pi) &= \pi(1-\pi)\pi\pi(1-\pi) \\ &= \pi^3(1-\pi)^2 \end{aligned} \tag{7}$$

Formule (7) kunnen we op twee manieren bekijken. We kunnen de uitkomst van het experiment als argument van de functie Prob bekijken en voor alle mogelijke uitkomsten van het experiment een uitdrukking vinden die analoog is aan het rechterlid van (7). Dan vinden we een aantal uitdrukkingen waarin π verschijnt als een vast (hoewel nog onbekend) getal. Daarom staat

π na de ';' in het linkerlid van (7). We kunnen (7) echter ook bekijken als een functie van π, waarbij we de uitkomsten van ons experiment beschouwen als een gegeven. Voor elke waarde van π die we dan invullen, krijgen we als uitkomst hoe waarschijnlijk onze observaties zijn, als π die waarde aanneemt. De functie (7) zo bekeken noemt men de aannemelijkheidsfunctie (Engels: Likelihood function) en die wordt gegeven door

$$L(\pi;(1\,0\,1\,1\,0)) = \text{Prob}((1\,0\,1\,1\,0);\pi). \qquad (8)$$

De grafiek van (8) is weergegeven in Figuur 5.



Figuur 5

Aannemelijkheidsfunctie
voor de observatie (1 0 1 1 0)

De ML-schatting van π is die waarde waarvoor de aannemlijkheidsfunctie zo groot mogelijk wordt, dat wil zeggen die waarde waarvoor de gegeven observaties de grootste waarschijnlijkheid hebben. In het voorbeeld is dit 0.6 zoals makkelijk uit Figuur 5 kan worden afgelezen. Natuurlijk zal men niet steeds een grafiek van de aannemelijkheidsfunctie maken om de schatting te bepalen. Met standaardtechnieken uit de wiskunde - die we hier niet bespreken - is het mogelijk af te leiden voor welke waarden van de parameter de aannemelijkheidsfunctie haar maximum bereikt. Stellen we door

s het aantal successen voor in het experiment en door n het aantal keren dat we opgooien, dan is de algemene uitkomst

$$\hat{\pi} = \frac{s}{n} \tag{9}$$

waarbij het dakje boven π aangeeft dat het om een schatter gaat. Het is bij dit voorbeeld nog belangrijk op te merken, dat alle experimenten met eenzelfde aantal successen dezelfde schatting van π opleveren, ongeacht hoe de afwisseling van successen en mislukkingen verloopt. Men drukt het wel eens als volgt uit: alle informatie die het experiment over π bevat, wordt uitgedrukt in het aantal successen. De fijnere structuur van de observaties (de preciese afwisseling van 1 en 0) kan ons geen extra informatie over π opleveren. Verderop zullen we zien dat die fijnere structuur voor andere doeleinden kan worden aangewend.

2.2 J.M.L schatting in het Rasch-model

We beginnen met een rechttoe rechtaan toepassing van de principes uit de vorige paragraaf in geval van het Rasch-model. De enige complicatie is dat we niet één maar meerdere parameters tegelijkertijd moeten schatten: namelijk, één voor elk item en één voor elke persoon bij wie de test is afgenomen. Bij k items en n personen moeten we dus n+k parameters schatten. (De J in J.M.L. staat voor 'joint' . Men gebruikt deze aanduiding niet omdat er meerdere parameters geschat worden, maar omdat de twee soorten parameters, persoonsparameters en itemparameters gezamenlijk geschat worden.) Om het principe duidelijk te maken beginnen we met n=1 en k=3. Er moeten dus vier parameters geschat worden: ϑ, σ_1, σ_2 en σ_3. Veronderstel dat we het antwoordpatroon (1 0 1) hebben geobserveerd, dan is duidelijk dat

$$L(\sigma_1,\sigma_2,\sigma_3,\vartheta;(101)) = f_1(\vartheta)\,(1-f_2(\vartheta))\,f_3(\vartheta). \tag{10a}$$

Let wel dat de vermenigvuldigingen in (10a) gerechtvaardigd worden door de veronderstelling van lokale stochastische onafhankelijkheid. Breiden we nu de observaties uit tot twee personen, zeg Jan en Piet, met vaardigheden ϑ_j

respectievelijk ϑ_p en antwoordpatronen (1 0 1) resp. (1 1 0), dan wordt de aannemelijkheidsfunctie gegeven door

$$L(\sigma_1,\sigma_2,\sigma_3,\vartheta_j,\vartheta_p;[(101),(110)]) = f_1(\vartheta_j)\,(1-f_2(\vartheta_j))\,f_3(\vartheta_j) \times f_1(\vartheta_p)\,f_2(\vartheta_p)\,(1-f_3(\vartheta_p)). \qquad (10b)$$

De vermenigvuldiging in (10b) die expliciet door het x-teken is aangegeven wordt gerechtvaardigd door de veronderstelling dat Jan en Piet onafhankelijk van elkaar antwoorden (dat wil zeggen dat ze niet van elkaar zitten over te schrijven). Deze vorm van onafhankelijkheid wordt wel eens experimentele onafhankelijkheid genoemd.

Het zal wel duidelijk zijn hoe de aannemelijkheidsfunctie opgeschreven dient te worden bij zeer veel observaties. (Het feit dat je dan een zeer lange uitdrukking vindt, is niet erg; er bestaan notationele conventies om zo'n uitdrukking kort op te schrijven, doch daarop gaan we hier niet in.) De grootste-aannemelijkheidsschattingen van de parameters zijn dan die waarden die de aannemelijkheidsfunctie zo groot mogelijk maken. In dit geval is het niet zo dat er een eenvoudige uitdrukking bestaat als formule (9); er is heel wat rekenwerk nodig om die waarden te vinden. Hoe dat precies gaat wordt hier niet uiteengezet.

Er is wel een ander probleem dat hier aan de orde gesteld moet worden. In het experiment met het muntstuk krijgen we informatie over π door observaties te verzamelen en hoe vaker we opgooien hoe meer informatie we krijgen. Het is dus redelijk te verwachten dat de nauwkeurigheid waarmee π wordt geschat, groter wordt naarmate n groter wordt. In de statistiek wordt bewezen dat de verhouding s/n (het relatieve aantal successen) zeer dicht bij de werkelijke parameter π komt te liggen, als n zeer groot wordt. Een schatter met die eigenschap wordt <u>consistent</u> genoemd. Bij het Rasch-model ligt de zaak echter ingewikkelder. Als we daar n laten toenemen, wordt het probleem steeds groter, want voor elke nieuwe observatie komt er een ϑ-parameter bij. Het gevolg hiervan is dat de σ-parameters niet consistent geschat worden. Dit is zo ernstig dat we de hierboven beschreven methode als onbruikbaar terzijde schuiven. In de twee volgende paragrafen wordt

beschreven hoe we het probleem van het steeds groeiend aantal ϑ-parameters kunnen omzeilen.

2.3 C.M.L. schatting in het Rasch-model

De eerste manier om de ϑ-parameters kwijt te raken bestaat erin de totale steekproef op te delen in homogene scoregroepen, dat wil zeggen in groepen personen die eenzelfde aantal items correct hebben, en de aannemelijkheid van een bepaald antwoordpatroon te bekijken binnen elke scoregroep afzonderlijk. Technisch drukt men dat uit door te zeggen dat men conditioneert op de score. (De C van CML staat voor conditional; in JML conditioneert men niet; in de literatuur wordt JML ook wel aangeduid als UML voor Unconditional Maximum Likelihood. We bekijken eerst een voorbeeld.

Veronderstel k=3 en we beschouwen het antwoord patroon (1 0 1). De score s van dit antwoordpatroon is 2. Nu zijn er exact drie mogelijke antwoordpatronen met score 2, nl. (1 0 1), (1 1 0) en (0 1 1). Conditioneren op score 2 betekent dat we reeds weten dat een van die drie antwoordpatronen is opgetreden, en nu willen we weten wat de kans is dat (1 0 1) is opgetreden, als alleen die drie mogelijk zijn. De formule hiervoor is

$$P(1\,0\,1|s=2,\vartheta) = \frac{P(1\,0\,1|\vartheta)}{P(1\,0\,1|\vartheta) + P(1\,1\,0|\vartheta) + P(0\,1\,1|\vartheta)}. \qquad (11)$$

Bekijken we nu even twee equivalente formules voor het Raschmodel:

$$Prob(X_i=1|\vartheta) = f_i(\vartheta) = \frac{e^{\vartheta-\sigma_i}}{1+e^{\vartheta-\sigma_i}}, \qquad (12a)$$

en

$$Prob(X_i=0|\vartheta) = 1 - f_i(\vartheta) = \frac{1}{1+e^{\vartheta-\sigma_i}}. \qquad (12b)$$

Als we de aannemelijkheidsfunctie opstellen moeten we producten nemen van uitdrukkingen met de gedaante (12a) voor juiste antwoorden of (12b) voor foute antwoorden. Bemerk dat de noemers van (12a) en (12b) identiek zijn. De noemer van het product is dus onafhankelijk van het specifieke antwoordpatroon. Stellen we deze noemer voor door het symbool K. Beschouw nu de probabiliteit op het antwoordpatroon (1 0 1):

$$\begin{aligned} \text{Prob}(1\,0\,1|\vartheta) &= \frac{e^{\vartheta}e^{-\sigma_1}e^{\vartheta}e^{-\sigma_3}}{K} \\ &= \frac{e^{2\vartheta}e^{-\sigma_1-\sigma_3}}{K}. \end{aligned} \tag{13}$$

In de teller van (13) komt 2ϑ voor in de exponent. Het is duidelijk dat die 2 daar staat, omdat het over een antwoordpatroon gaat met precies 2 juiste antwoorden. Doch dit is ook het geval voor de antwoordpatronen (1 1 0) en (0 1 1). Dan is het niet moeilijk in te zien dat

$$\begin{aligned} P(1\,0\,1|s=2,\vartheta) &= \frac{\dfrac{e^{2\vartheta}e^{-\sigma_1-\sigma_3}}{K}}{\dfrac{e^{2\vartheta}e^{-\sigma_1-\sigma_3}}{K}+\dfrac{e^{2\vartheta}e^{-\sigma_1-\sigma_2}}{K}+\dfrac{e^{2\vartheta}e^{-\sigma_2-\sigma_3}}{K}} \\ &= \frac{e^{-\sigma_1-\sigma_3}}{e^{-\sigma_1-\sigma_3}+e^{-\sigma_1-\sigma_2}+e^{-\sigma_2-\sigma_3}} \end{aligned} \tag{14}$$

Het belangrijke aspect van (14) is dat het rechterlid onafhankelijk is van ϑ en alleen nog een functie van de itemparameters. Bij de vereenvoudiging van (14), dat wil zeggen de overgang van het tweede lid naar het derde lid, merken we dat niet alleen de noemers K verdwijnen, maar ook de uitdrukking 2ϑ. Dit kon alleen maar door ervoor te zorgen dat ϑ telkens met hetzelfde getal (2) werd vermenigvuldigd. Maar 2 is precies de score die met de drie beschouwde antwoordpatronen is geassocieerd. De 'truc' om ϑ te laten verdwijnen werkt dus alleen maar, als we conditioneren op de score.

De uitdrukking (14), maar nu beschouwd als een functie van de σ-parameters, noemen we de conditionele aannemelijkheidsfunctie (voor één patroon,

namelijk (1 0 1).). De conditionele aannemelijkheidsfunctie voor alle geobserveerde antwoordpatronen samen is gewoon het product (waarom ?) van soortgelijke uitdrukkingen. De conditionele grootste-aannemelijkheids-schattingen zijn dan die waarden van de σ-parameters die dit product zo groot mogelijk maken. Het vinden van die waarden is behoorlijk ingewikkeld, doch daar draagt het computerprogramma zorg voor.

Deze methode van schatten heeft twee grote voordelen:

(1) Het is bewezen dat deze manier van schatten consistente schatters oplevert die bovendien zo nauwkeurig mogelijk zijn. (Op de preciese betekenis van 'zo nauwkeurig mogelijk' kunnen we hier niet ingaan. Het betekent zoiets als: er wordt door deze methode eigenlijk geen informatie over de σ-parameters weggegooid.)

(2) Tot hiertoe hebben we nog niets gezegd over de manier waarop de geteste personen uit de populatie getrokken dienen te worden. Dit is met opzet gebeurd. Er is niet stilzwijgend verondersteld dat de steekproef een aselekte trekking moet zijn uit de populatie. Integendeel, door gebruik te maken van de CML-methode maakt het in principe niets uit hoe de steekproef uit de populatie is getrokken. Immers de CML-methode wordt gebruikt om iets te kunnen zeggen over de itemparameters en niet over de populatie van personen. Bij de derde schattingsmethode, die in de volgende paragraaf wordt besproken, hebben we dit voordeel niet.

2.4 M.M.L. schatting in het Rasch-model

Een tweede methode om de individuele ϑ-parameters kwijt te raken bestaat erin ze een andere status te geven. De status van de ϑ-waarden is het standpunt van waaruit men de gegevens beschouwt. Tot nog toe hebben we eigenlijk impliciet aangenomen dat, als Jan en Piet in de steekproef zitten, we terzelfdertijd geïnteresseerd zijn in de waarde van de itemparameters en in de ϑ-waarde van Jan en Piet (en van precies al die personen die in de steekproef zitten.) Een ander standpunt is dat het ons

eigenlijk niet kan schelen wie in de steekproef zit, omdat we toch alleen maar geïnteresseerd zijn in de itemparameters. Dit impliceert dat we de steekproef als een aselecte steekproef uit een of andere populatie beschouwen en dat we de gedragingen van die toevallige steekproef willen gebruiken om de itemparameters te schatten. Dit standpunt biedt de mogelijkheid om ϑ kwijt te raken op de volgende manier.

Veronderstel dat ϑ slechts 3 verschillende waarden kan aannemen in de populaties, namelijk -1, 0 en 1, en veronderstel dat deze waarden in de populatie voorkomen met een proportie van resp. .25 .35 en .4. We beschouwen nu de kans dat we het antwoordpatroon $\underline{x}$ = (1 0 1) observeren bij aselecte trekking van een persoon uit de populatie. Deze kans is gegeven door

$$P(\underline{x}) = P(\underline{x}|\vartheta=-1)\,0.25 + P(\underline{x}|\vartheta=0)\,0.35 + P(\underline{x}|\vartheta=1)\,0.40 \qquad (15)$$

Dat wil zeggen, als we ϑ niet kennen, kunnen we alle conditionele kansen $P(\underline{x}|\vartheta)$ als het ware gaan middelen door te vermenigvuldigen met de kans dat die ϑ optreedt, en die gewogen conditionele kansen op te tellen. Het resultaat noemt men niet de gemiddelde kans, maar de marginale kans. (Vandaar de eerste M in MML.) Laten we dit nu veralgemenen tot de situatie waar ϑ W verschillende waarden kan aannemen:

$$\text{Prob}(\underline{x}) = \sum_{i=1}^{W} \text{Prob}(\underline{x}|\vartheta_i)\,\text{Prob}(\vartheta_i)\,. \qquad (16)$$

Het gebruik van (16) zonder meer is niet erg aantrekkelijk, omdat we dan een waarde voor W moeten kennen, de verschillende waarden die ϑ kan aannemen en de kansen $\text{Prob}(\vartheta_i)$. Als we die niet kennen, moeten we ze ook uit de data gaan schatten, zodat er naast de itemparameters nog eens 2W parameters bijkomen. Hoe paradoxaal het ook klinkt, de uitweg uit dit probleem bestaat erin ϑ oneindig veel waarden te laten aannemen (en nog sterker: te veronderstellen dat ϑ continue is) en een bepaalde regel te

veronderstellen waaruit de 'kans' op een bepaalde ϑ uit ϑ zelf kan bepaald worden. (We mogen bij continue variabelen niet meer spreken van 'kans'; men spreekt van dichtheid. Die dichtheid duiden we aan met het functiesymbool g.) We kennen een heel populaire dichtheid, nl. die van de normale verdeling:

$$g(\vartheta) = \frac{1}{\sqrt{2\pi\tau^2}} e^{-\frac{(\vartheta-\mu)^2}{2\tau^2}} \tag{17}$$

waarin $\pi = 3.14159...$ We zien dat in die functieregel twee parameters voorkomen, nl μ en τ^2, het gemiddelde en de variantie van ϑ. De marginale probabiliteit van antwoordpatroon $\underline{x}$ in het geval we een normale verdeling veronderstellen van ϑ, is dan gegeven door

$$\begin{aligned} \text{Prob}(\underline{x}) &= \int_{-\infty}^{+\infty} \text{Prob}(\underline{x}|\vartheta)\, g(\vartheta)\, d\vartheta \\ &= \int_{-\infty}^{+\infty} \text{Prob}(\underline{x}|\vartheta)\, \frac{1}{\sqrt{2\pi\tau^2}} e^{-\frac{(\vartheta-\mu)^2}{2\tau^2}}\, d\vartheta \end{aligned} \tag{18}$$

Formule (18) is niet meer afhankelijk van ϑ, wel van de itemparameters en van de twee verdelingsparameters μ en τ^2. Indien we deze marginale probabiliteit nu beschouwen als functie van die parameters, dan krijgen we de aannemelijkheidsfunctie voor het antwoordpatroon $\underline{x}$. De aannemelijkheidsfunctie voor alle geobserveerde antwoordpatronen samen is dan niets anders dan het product van soortgelijke functies als (18), waarin $\underline{x}$ staat als algemeen symbool voor de geobserveerde antwoordpatronen. In de statistiek wordt bewezen dat door deze methode consistente schattingen van alle parameters worden verkregen.

We sluiten deze paragraaf af met een korte vergelijking van de CML- en de MML-methode.

Het belangrijkste verschil tussen beide methodes bestaat erin dat bij CML geen enkele veronderstelling wordt gemaakt over de verdeling van ϑ in de populatie, terwijl dat bij MML wel wordt gedaan. Het is bij MML helemaal niet noodzakelijk een normale verdeling te veronderstellen; men zou ook een andere verdeling kunnen veronderstellen. Belangrijk is echter in te zien dat de veronderstelling over de verdeling nu deel gaat uitmaken van het model. Dus als we MML toepassen (met de normale verdeling) dan vermengen we als het ware twee modellen: het Raschmodel dat iets vertelt over de antwoorden gegeven ϑ, en de normale verdeling die vertelt hoe de ϑ's in de populatie zijn verdeeld. De verstrengeling van beide modellen gebeurt op een heel diep niveau (zie formule (18)), zodanig dat beide onderdelen niet eenvoudig uit elkaar zijn te halen. Maken we een fout in de veronderstelling over de normale verdeling (ϑ is niet normaal verdeeld, dan wel ϑ is normaal verdeeld, maar onze steekproef is niet aselect uit deze verdeling getrokken), dan heeft dat als gevolg dat er ook systematische fouten geïntroduceerd worden in de schatting van de itemparameters. Een gebruiker die MML gebruikt stelt zich dus iets kwetsbaarder op.

Het voordeel van MML is dan wel dat de verdelingsparameters gelijktijdig met de itemparameters geschat kunnen worden. Indien men in beide geïnteresseerd is, is MML de meest efficiente methode.

2.5 Het aantal vrije parameters in het Rasch-model

Tot nog toe hebben we gedaan alsof we bij een toets van k items ook k parameters moeten schatten. Dit is echter niet helemaal juist. Uit formules (3) en (4) weten we dat de itemresponsfunctie in het Rasch-model de logistische functie is met als argument $\vartheta - \sigma_1$. Als dit verschil vastligt, ligt de functiewaarde vast en als de functiewaarde vastligt ligt het verschil ook vast. Doch als het verschil $\vartheta-\sigma_1$ vastligt, betekent dit niet dat ϑ en σ_1 allebei vastliggen. Definieer

$$\vartheta^* = \vartheta + b$$
$$\sigma^* = \sigma + b,$$

waarbij b een willekeurig getal is, dan is duidelijk dat $\vartheta^*-\sigma^* = \vartheta-\sigma$. Als we dus een stelletje σ's geschat hebben, kunnen we een evenwaardig stelletje vinden door bij de eerste een willekeurig getal op te tellen. Omdat we dit getal vrij mogen kiezen, kunnen we het zo kiezen dat bijvoorbeeld $\sigma_1 = 0$. Maar als we één parameter gelijkstellen aan 0, dan hoeft hij niet meer geschat te worden. Dus worden er maar k-1 parameters geschat. Het vastleggen van een parameter staat gelijk met het uitkiezen van een oorsprong van de schaal. Dit vastleggen van de oorsprong noemt men normalisatie. In het programma OPLM wordt de oorsprong gelijkgesteld aan het gemiddelde van de parameterschattingen. Indien de MML methode wordt gebruikt, dienen er naast de k-1 vrije itemparameters ook nog twee verdelingsparameters geschat te worden. Dus in MML zijn er k+1 vrije parameters. Soms gebeurt het dat men in MML de oorsprong laat samenvallen met het gemiddelde van de populatie. In dat geval kan men niet nog eens een itemparameter vrij kiezen. Dan zijn er dus k vrije itemparameters en 1 vrije verdelingsparameter (de variantie τ^2), in totaal k+1.

2.6 De eenheid van de schaal

In de vorige paragraaf hebben we gezien dat we de oorsprong van de schaal vrij kunnen kiezen. Dit is ook zo voor de eenheid van de schaal, hoewel in de literatuur vaak beweerd wordt dat dit niet zo is. Veronderstel dat we de volgende transformatie doorvoeren:

$$\alpha\vartheta' = \vartheta$$
$$\alpha\sigma' = \sigma$$

met $\alpha > 0$. Het verschil $(\vartheta-\sigma)$ kunnen we dus vervangen door $\alpha(\vartheta'-\sigma')$. Het Rasch-model kunnen we dan iets algemener opschrijven als

$$f_i(\vartheta) = \frac{e^{\alpha(\vartheta-\sigma_i)}}{1+e^{\alpha(\vartheta-\sigma_i)}}. \qquad (19)$$

Voor α mogen we een willekeurig positief getal kiezen. Het is de gewoonte hiervoor de waarde 1 te kiezen, doch dit is absoluut niet noodzakelijk. Bemerk verder in formule (19) dat α geen index heeft; α mag dus niet per persoon en/of per item een andere waarde hebben, maar is een constante. Verderop zullen we zien dat we een veel krachtiger model krijgen door die α van item tot item een andere waarde te laten aannemen. Voorlopig kunnen we ermee volstaan te constateren dat in het Raschmodel α constant hoort te zijn.

3 Het programmapakket OPLM (I)

Het programmapakket OPLM bevat een aantal programma's waarmee data kunnen worden geanalyseerd volgens het Rasch-model en volgens enkele veralgemeningen van het Rasch-model, die de naam 'One Parameter Logistic Model' hebben gekregen. Deze veralgemeningen komen in latere hoofdstukken aan de orde. Dit hoofdstuk bevat een korte en gedeeltelijke handleiding die de gebruiker in staat stelt schattingen van de parameters in het Raschmodel te laten berekenen

3.1 De structuur van het gegevensbestand

Het programmapakket OPLM is geschikt voor de analyse van onvolledige data. Onvolledige data kunnen op twee manieren ontstaan. Bij het beantwoorden van items kan een leerling (om welke reden dan ook) bepaalde items overslaan of er niet aan toekomen. Op die manier ontstaan incidentele 'gaten' in de observaties. Het kan ook zijn dat men bijvoorbeeld antwoorden wil verzamelen op 100 items, waarbij het als onredelijk wordt beschouwd elke leerling alle 100 items te laten beantwoorden. Daarom geeft men aan elke leerling slechts een gedeelte van de items. Op die manier zijn de gaten in de observaties gepland, en men spreekt van 'structureel onvolledige data'. OPLM is alleen bedoeld voor structureel onvolledige data. Overgeslagen of niet bereikte items (die op een speciale manier gecodeerd zijn, bijvoorbeeld als '8' of '9') worden automatisch geïnterpreteerd als verkeerde antwoorden. Het is mogelijk het programma te dwingen incidentele gaten als structurele te beschouwen. Daar dient men echter zeer voorzichtig mee te zijn. Het 'techniekje' zal wel werken, doch de conclusies die men uit zulk een analyse haalt zijn vaak niet gerechtvaardigd.

Het begrip 'boekje' (booklet)

Een boekje is een deelverzameling van het totaal aantal items dat men wil analyseren. Deze deelverzameling wordt aan een deel van de totale steekproef ter beantwoording aangeboden. We zullen de organisatie van het gegevensbestand bespreken aan de hand van het volgende voorbeeld. Veronderstel dat men in totaal 10 items wil analyseren. De totale

steekproef leerlingen is opgedeeld in twee groepen. Groep 1 krijgt de items 1 t/m 7 aangeboden; groep 2 krijgt de items 5 t/m 10 aangeboden. Hier is dus sprake van twee boekjes: boekje 1 = (1,2,3,4,5,6,7) en boekje 2 = (5,6,7,8,9,10). De antwoorden van alle leerlingen moeten ondergebracht worden in één bestand. Om het programma goed te laten werken, zullen we ook moeten meedelen welk boekje elke leerling heeft gekregen. In dit voorbeeld zullen we afspreken dat het boekjesnummer opgenomen wordt als een tweecijferig getal en dat dit getal de kolommen 1 en 2 van elk record zal beslaan. Verder spreken we af dat de itemscore bestaat uit ééncijferige getallen die aaneengesloten op het gegevensbestand staan, beginnend in kolom 11. De kolommen 3 tot 10 zijn voorbehouden voor leerling-identificatie. De gegevens kunnen nu op twee manieren opgenomen worden in het gegevensbestand: 'extended' en 'compressed'. Een gedeelte van het gegevensbestand is afgebeeld in Figuur 6 met de structuur 'extended'. De ontbrekende data zijn als '9' gecodeerd. Indien men de gegevens op deze manier op het scherm laat verschijnen, heeft dit twee voordelen: het patroon van de 9's geeft duidelijk aan om welk boekje het gaat en bovendien verwijst elke kolom in het gegevensbestand naar hetzelfde item. Het nadeel is natuurlijk dat men in het bestand plaats moet inruimen voor de niet geobserveerde itemantwoorden. Bij een totaal van 250 items waarbij elke leerling 50 items heeft beantwoord, bestaat het gegevensbestand voor ongeveer 80% uit non-informatie. In zo'n geval is het handiger om de 'compressed' structuur te gebruiken.

```
0        1         2   ...
12345678901234567890123...
 1    wim 1011011999
 1   anne 0110110999
 .     .       .
 .     .       .
 1   kees 1001111999
 2   hans 9999110111
 2 louise 9999110010
 .     .       .
 .     .       .
```

Figuur 6

Gegevensbestand met 'extended' strucuur

Deze structuur kan voor de gegevens in Figuur 6 als volgt geconstrueerd worden: verwijder alle codes voor ontbrekende data, en schuif de overblijvende gegevens naar links. Dit is afgebeeld in Figuur 7. Bemerk dat niet alle records even lang zijn, want de twee boekjes bevatten niet evenveel items. Bovendien verwijst een bepaalde kolom niet meer steeds naar het zelfde item. In het record van Anne verwijst kolom 11 naar item 1, doch in het record van Louise verwijst kolom 11 naar item 5. Een item wordt in dit geval dus geïdentificeerd door het boekjesnummer en de kolom. Verderop zullen we zien dat deze informatie zeer gemakkelijk is mee te delen aan het programma.

```
0        1         2   ...
12345678901234567890123...
 1    wim 1011011
 1   anne 0110110
 .      .       .
 .      .       .
 1   kees 1001111
 2   hans 110111
 2 louise 110010
 .      .       .
 .      .       .
```

Figuur 7

Gegevensbestand met 'compressed' strucuur

Voor het gegevensbestand dient men volgende regels in acht te nemen:

(1) De itemantwoorden moeten voor elk record in dezelfde kolom beginnen.

(2) Bij elk record moet een boekjesnummer worden opgenomen, ook bij zogenaamde volledige designs (dat wil zeggen designs met één boekje). Deze boekjesnummers moeten in elk record op dezelfde plaats staan. (Bij meer dan 9 boekjes dienen er dus minimaal twee kolommen gereserveerd te worden voor elk boekje.)

(3) Boekjesnummers moeten positieve gehele getallen zijn. (Ze hoeven niet bij 1 te beginnen en elkaar niet in stappen van 1 op te volgen, doch

dit brengt extra werk mee bij het opgeven van de analysespecificatie (zie paragraaf 3.3.3).

(4) De 'itemgegevens' zijn itemscores, dat wil zeggen het aantal punten dat op elk item is behaald. Deze scores worden als gehele getallen ingelezen (geen decimale punt). De itemgegevens zijn dus niet de gekozen alternatieven bij meerkeuzevragen; het programma versleutelt niet.

(5) Bij de 'compressed' structuur dient men ervoor te zorgen dat alle boekjes in hetzelfde formaat staan. Voorbeeld: in Figuur 7 verwijzen de kolommen 11 en 12 naar de items 1 en 2 bij boekje 1. Bij boekje 2 moeten dezelfde kolommen ook naar twee items verwijzen (en niet bijvoorbeeld naar één tweecijferig itemantwoord). Men kan het veiligst werken door aan alle itemantwoorden hetzelfde aantal kolommen toe te kennen.

(6) Alle gegevens van één persoon moeten in één record staan. Dus niet meerdere records per persoon.

(7) De volgorde van de records in het gegevensbestand is willekeurig. Het is echter zeer aan te bevelen records uit hetzelfde boekje bij elkaar te plaatsen. Bij volledig willekeurige volgorde kan de inleestijd van het gegevensbestand gemakkelijk toenemen met een factor 100. De volgorde van de boekjes in het gegevensbestand is willekeurig, en ordening levert geen tijdswinst op.

(8) Gebruikt men de 'extended' structuur, dan moet ook een numerieke code worden opgegeven voor de niet aangeboden items. Deze code is willekeurig.

(9) Een spatie ('blank') wordt door het programma geïnterpreteerd als een nul.

(10) Het programma is ook geschikt voor polytome items, dit zijn items waarop men niet uitsluitend 0 of 1 kan behalen maar bijvoorbeeld 0, 1,

2, 3 of 4 punten. Indien een item wel is aangeboden maar overgeslagen of niet bereikt wordt staat in het data bestand soms een speciale code (bij voorbeeld '9') die niet behoort tot de mogelijke itemscores. Het programma leest deze codes zonder leesfout, doch interpreteert ze als een itemantwoord met score 0.

3.2 Algemene structuur van OPLM

Figuur 8 is het algemene stroomdiagram van het programmapakket. De dubbel omlijste namen zijn programma's die zelfstandig kunnen worden uitgevoerd. De enkelvoudig omlijste namen zijn namen van bestanden die ofwel voorhanden dienen te zijn (het gegevensbestand) of die door een van de programma's worden aangemaakt (alle andere bestanden). Sommige van die bestanden zijn zgn. ASCII-bestanden (aangeduid met ('(a)'). Dit zijn tekstbestanden die zonder meer naar de printer kunnen worden gestuurd of op het scherm zichtbaar gemaakt met een editor. Andere bestanden zijn zogenaamde binaire bestanden (aangeduid met '(b)'), die alleen door andere programma-onderdelen kunnen worden gelezen. Het printen van zulke bestanden heeft geen zin.

Alle bestanden die door het programma worden aangemaakt, hebben dezelfde naam, maar een verschillende extensie waaruit kan afgeleid worden welk programma-onderdeel ze heeft aangemaakt en welk soort informatie ze bevatten. Deze naam wordt in het stroomdiagram aangegeven met het generieke symbool JOBN. De specifieke naam die deze bestanden zullen dragen wordt door de gebruiker zelf gekozen (zie verder). Het gegevensbestand heeft een willekeurige naam. In het stroomdiagram is de naam 'DATA' gekozen.

3.2.1 Algemene beschrijving van de programma's

Het programma OPIN is een interactief programma waarmee de gebruiker de specificaties opgeeft over de uit te voeren analyse, zoals welk soort model moet worden gebruikt, waar de gegevens te vinden zijn, hoe ze moeten worden gelezen, hoe de files moeten heten, enz. Als de specificatie is opgegeven,

kan ze weggeschreven worden in het bestand JOBN.SCR. Het opgeven van die specifikaties kan gebeuren 'from scratch', doch men kan ook een bestaande .SCR file binnenhalen, en de bestaande specificaties veranderen.

Het programma OPPRE doet een soort vooranalyse. Omdat verschillende gebruikers hun data op zeer verschillende manieren in het databestand kunnen zetten, is het handig een soort gestandaardiseerd databestand aan te maken, dat van het oorspronkelijke databestand alleen die informatie bevat die nodig is voor de analyse. Dit gestandaardiseerde bestand heet JOBN.PRE. Naast de data bevat dit bestand ook in een speciaal formaat alle specificaties die de gebruiker via het programma OPIN heeft opgegeven. Om de eigenlijke analyse te kunnen uitvoeren zijn DATA-bestand en het .SCR bestand dan verder niet meer nodig. Bovendien bevat het .PRE bestand voldoende informatie om de parameters te kunnen schatten met de CML methode of met de MML methode. Met het programmapakket OPLM kan men ook een eenvoudige toets- en item-analyse laten uitvoeren volgens de klassieke testtheorie. Deze analyse wordt uitgevoerd door het programma OPPRE. De resultaten worden eveneens in het .PRE-bestand weggeschreven.

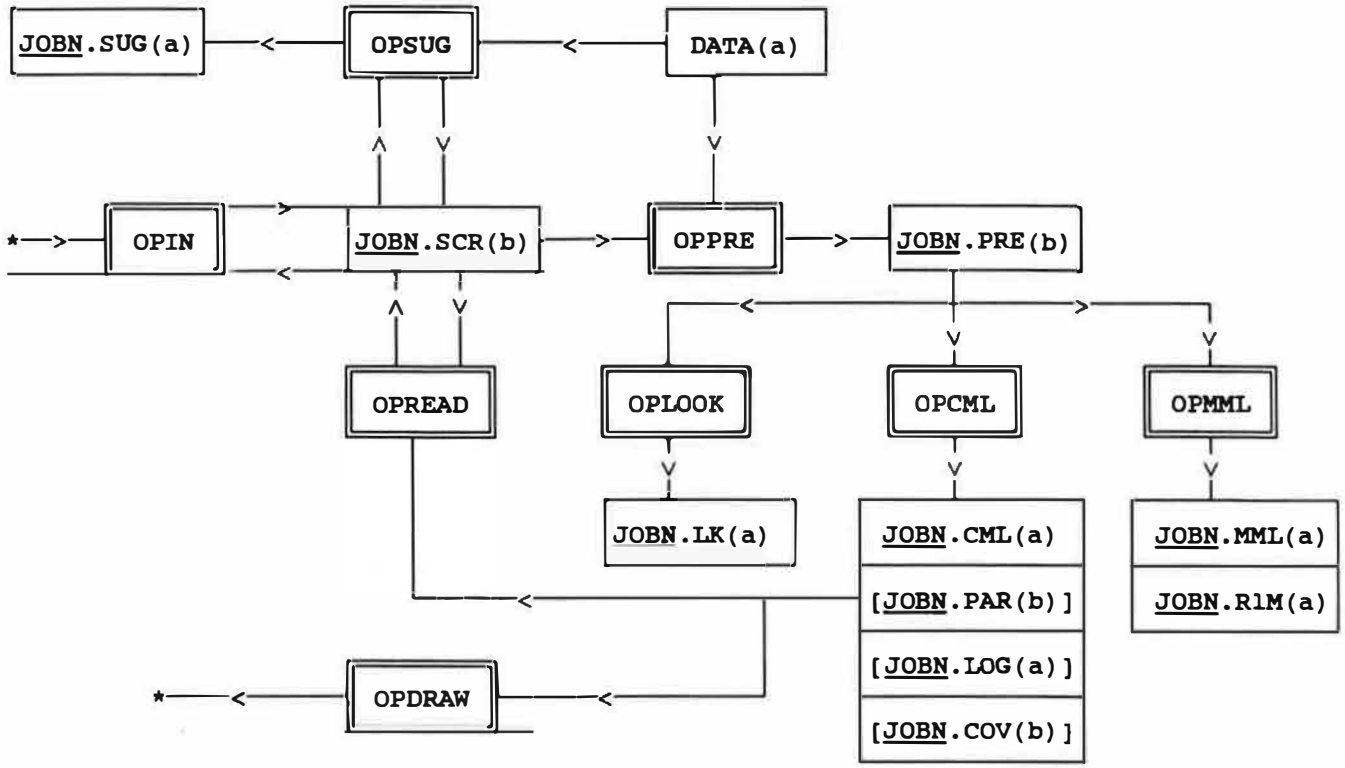
Het programma OPCML schat de parameters van het model uit de data volgens de CML methode. Daarnaast voert het programma ook statistische toetsen uit die toelaten te beoordelen of het model aanvaardbaar is. Deze toetsen komen in het volgende hoofdstuk ter sprake. Het programma schrijft de resultaten van de analyse inclusief de resultaten van de klassieke toets- en item-analyse in het bestand-JOBN.CML. Bij de specificatie van de analyse (zie OPIN) kan men ook vragen nog drie andere bestanden aan te maken.

Het bestand-JOBN.PAR bevat de informatie die nodig is om grafische output te kunnen aanmaken met het programma OPDRAW.

Tijdens het uitvoeren van de analyse schrijft het programma boodschappen naar het scherm die de gebruiker informeren over de voortgang van de analyse, of die aangeven dat er iets fout gaat. Deze informatie kan men ook naar een speciaal bestand laten wegschrijven. Dit bestand is JOBN.LOG

One Parameter Logistic Model

Algemeen stroomdiagram



* = toetsenbord/scherm
(a) = ASCII bestand
(b) = binair bestand
[ ] = optioneel

Figuur 8

Stroomdiagram OPLM

Het bestand-JOBN.COV is bedoeld voor toekomstige uitbreidingen van het programmapakket. Nu heeft het nog geen zin de aanmaak van dit bestand te vragen.

Het programma OPMML schat de parameters van het model uit de data volgens de MML-methode (inclusief statistische toetsen). De resultaten worden geschreven in het bestand-JOBN.MML. Resultaten die alleen de statistische toetsen betreffen worden geschreven in het bestand-JOBN.R1M. De resultaten van een MML analyse hebben (nog) geen grafische weergave.

Het programma OPLOOK. Indien men bij de specificatie van de opdracht fouten maakt (wensen anders opgeeft dan bedoeld), is de kans groot dat programma's OPCML of OPMML vastlopen en eindigen met een niet al te doorzichtige foutmelding. In zulke gevallen kan het nuttig zijn het .PRE-bestand te inspecteren. Het programma OPLOOK zet het binaire .PRE-bestand om in een leesbaar ASCII-bestand, dat op het scherm kan worden geïnspecteerd. De gegevens die op het .PRE-bestand staan zijn dan eveneens voorzien van enig commentaar. Dit bestand draagt de extensie .LK. Indien een klassieke toets- en item-analyse is gevraagd, staat het resultaat van deze analyse ook in het .LK-bestand.

De programma's OPREAD en OPSUG worden in hoofdstuk 6 besproken.
Het programma OPDRAW wordt in hoofdstuk 8 besproken.

3.2.2 Het aanroepen van de programma's

Men start het programma OPIN door het intikken van OPIN [CR]. Verder is het programma menugestuurd.

De andere programma's (met uitzondering van OPREAD) kunnen op twee manieren gestart worden. We illustreren dit met het programma OPPRE.

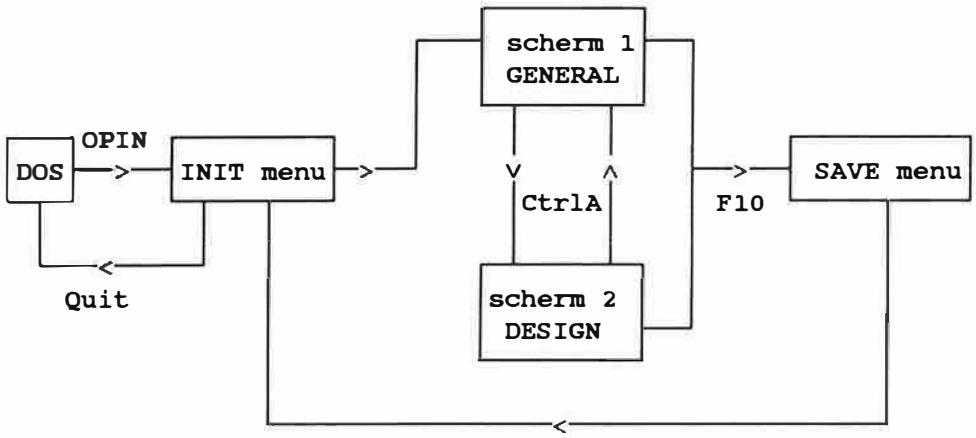
(a) Het commando OPPRE JOBN [CR] start het programma met het .SCR bestand JOBN.SCR.

(b) Het commando OPPRE [CR] start het programma met het .SCR-bestand dat het laatst door OPIN is aangemaakt c.q. ingelezen. Dit gebruik vergt enige oplettendheid. Stel dat het .SCR-bestand EXAMPLE1.SCR bestaat. Nu wordt met OPIN het bestand EXAMPLE2.SCR aangemaakt. Alvorens te beginnen met de analyse worden de specificaties van EXAMPLE1 nog even bekeken door ze via OPIN op het scherm te halen. Het commando OPPRE [CR] gebruikt dan automatisch EXAMPLE1.SCR. Wil men dat EXAMPLE2.SCR gebruikt wordt, dan dient te worden ingetikt OPPRE EXAMPLE2 [CR].

Het programma OPREAD wordt opgestart met het commando OPREAD [CR]. Verdere informatie wordt door het programma zelf opgevraagd.

3.3 Het programma OPIN

Het programma bevat twee menu's (het INIT-menu en het SAVE-menu) en twee schermen (GENERAL en DESIGN) die door de gebruiker kunnen worden ingevuld c.q. gewijzigd. De manier waarop men door die menu's en schermen loopt, is weergegeven in Figuur 9.

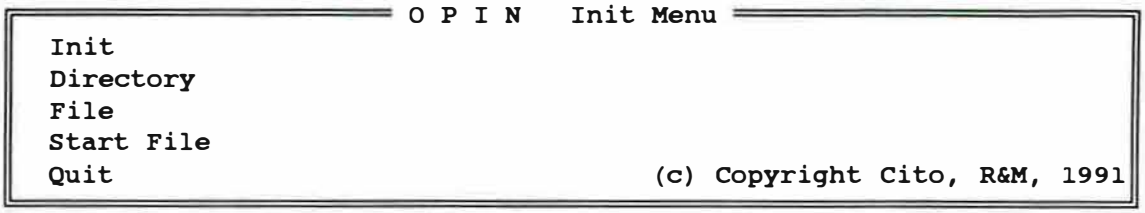


Figuur 9

Stroomdiagram van OPIN

3.3.1 Het INIT-menu

Figuur 10 is een afbeelding van het INIT-menu. Een keuze wordt gemaakt door ofwel met de cursor op het desbetreffende veld te gaan staan en op [CR] te drukken, ofwel door de eerste letter van de optie in te tikken.



Figuur 10

Het INIT-menu

Init. Bij deze keuze moeten de schermen helemaal worden ingevuld. Deze optie zal niet veel gebruikt worden, omdat het meestal makkelijker is bestaande schermen te wijzigen.

Directory. Indien men een .SCR-file wil inlezen, dient de directory waar de file zich bevindt te worden opgegeven. Geeft men niets op dan wordt alleen gezocht in de actieve directory. (Dat wil zeggen de directory 'vanwaar men werkt'.)

File. Hier kan men de naam intikken van het .SCR-bestand dat men wil wijzigen. Door het intikken van *.* krijgt men een lijst te zien van alle .SCR bestanden in de opgegeven directory. Hieruit kan men kiezen.

Start File. Dit veld heeft ongeveer dezelfde functie als File. Het is handig om hier een bestandsnaam in te vullen van een .SCR-bestand dat men steeds als vertrekpunt wil nemen voor wijzigingen. Men kan een bestand ook als start-file installeren door het als dusdanig weg te schrijven in het Save-menu. (Zie paragraaf 3.3.4.)

Quit. Terug naar DOS.

Als men kiest voor Init of na het invullen van een bestandsnaam voor File of Start File, komt men automatisch in scherm 1 (GENERAL) terecht.

3.3.2 Scherm 1: GENERAL

In Figuur 11 is scherm 1 afgebeeld, zoals men het te zien krijgt bij de optie Init in het INIT-menu. Het sterretje (*) geeft de plaats van de cursor aan. Op de onderste regel van het scherm verschijnt algemene informatie. Links staan de toegelaten waarden die men mag invullen op de plaats waar de cursor staat. Rechts staat wat men moet doen om hulp te krijgen. De volgende conventies worden in acht genomen:

^ betekent 'houd de Ctrl-toets ingedrukt';
@ betekent 'houd de Alt-toets ingedrukt'.

^F1 geeft informatie over hoe het invullen van het scherm precies verloopt. Itemlabels kunnen bijvoorbeeld van een afzonderlijk bestand worden ingelezen (met @F8).

^F2 geeft informatie over de betekenis van het veld waar de cursor staat.

De velden zelf zijn aangeduid met een naam die precies één hoofdletter bevat. Men kan de cursor naar dit veld laten springen door die letter in te drukken in combinatie met de Alt-toets. Bijvoorbeeld: @T laat de cursor naar het veld Title springen. In sommige gevallen zal men merken dat deze willekeurige sprongen niet zomaar uitgevoerd worden, doch dat in plaats daarvan een waarschuwing op het scherm verschijnt. Dit komt omdat het programma zeer sterk beveiligd is tegen inconsistenties. Het programma laat niet toe dat de gebruiker zichzelf tegenspreekt of essentiële informatie achterhoudt. Het verdient aanbeveling de waarschuwingen die het programma geeft, serieus te nemen.

#Items. Het totale aantal items dat in de studie betrokken is. In het voorbeeld van paragraaf 3.1 is dit aantal 10. Het verdient aanbeveling dit aantal niet te veranderen, als bijvoorbeeld blijkt dat er een 'slecht' item

tussen zit. (Zo'n item kan 'uitgezet' worden; zie onder cAl.) Door voor verschillende analyses dit aantal constant te houden verwijst het itemnummer (veld: Nr) steeds naar hetzelfde item. Het veld Nr kan dan dus geïnterpreteerd worden als het 'absolute' itemnummer. Dit veld is dan ook met de cursor niet bereikbaar.

```
= = = = = = = = = = = = = = O  P  L  M = = = = = = = = = = = = = =
= = = = = = = = = = = = = = G e n e r a l = = = = = = = = = = = = = =

═══ Analysis ═══                    #Items *···
Classical                          Nr Label      cAl  Disc  maXs  paR(s)
Pers pars                           1 Item   1   On     1     1   Free
Est  proc                           2 Item   2   On     1     1   Free
item tSts                           3 Item   3   On     1     1   Free
                                    4 Item   4   On     1     1   Free
═══ OutPut ═══                      5 Item   5   On     1     1   Free
Name  :          siZe   Standard    6 Item   6   On     1     1   Free
loG     Yes      taBles 2 columns   7 Item   7   On     1     1   Free
paraM   Yes                         8 Item   8   On     1     1   Free
coVar   No

Title :
Komment                                              ═══ Alt F9 ═══
                                                     General   Design
═══ Data ═══
direQt:
fYle  :                 strUcture
Format: ()

Range: 1 - 1000                                      ^F1-Info ^F2-Help
```

Figuur 11

Scherm 1 (GENERAL) bij keuze Init

Label. Een willekeurig label (van maximaal 8 tekens) waarmee het item geïdentificeerd wordt. Label en absoluut itemnummer verschijnen steeds samen in de uitvoer.

cAl. (On/Off) On betekent dat het item meedoet in de analyse; Off betekent dat het item niet meedoet. In dat geval wordt het itemantwoord wel van het databestand gelezen, doch verder wordt er niets mee gedaan.

Disc. (Zie ook hoofdstuk 4). Dit is de waarde van α uit formule (19). Wil men met het Rasch-model werken, dan dient Disc voor elk item dezelfde waarde te hebben. (De waarde van Disc heeft wel invloed op de berekeningen: hoe groter de hier ingevulde waarden, hoe langer de rekentijd en hoe groter het geheugenbeslag.)

maXs. (Zie ook hoofdstuk 6 en 7.) Dit is het maximale aantal punten dat iemand op een item kan behalen. Indien men het Raschmodel toepast is dit maximum 1.

paRs. (Zie hoofdstuk 6.)

Classical. (Yes/No) Voer een klassieke toets- en item-analyse uit.

Pers pars. (Yes/No) Bereken na het schatten van de itemparameters ook een schatting van de ϑ-waarde voor elke persoon in de steekproef. (Zie hoofdstuk 6 voor een bespreking hiervan.)

Est proc. (=Estimation procedure)(Complete/Diagonal). Zowel OPCML als OPMML gebruiken twee verschillende manieren om de parameters te schatten. De schattingen zijn in beide gevallen identiek. De verschillen kunnen als volgt worden omschreven:

> Complete: gebruik een methode die snel convergeert (d.w.z. die weinig stappen nodig heeft); nadeel: elke stap is duur in tijd; voordeel: de schatting van de standaardfout is nauwkeuriger. (Technisch: het Newton-Raphsonalgoritme wordt gebruikt. Voor het berekenen van de standaardfout wordt de informatiematrix gebruikt.)
>
> Diagonal: gebruik een methode die traag convergeert, doch waarbij elke stap snel verloopt. Nadeel: de schatting van de standaardfout is minder nauwkeurig. (Technisch: de schattingsmethode die gebruikt wordt staat in Fischer (Einführung in die Theorie psychologischer Tests, 1974, pp. 241-245); voor de schatting van de standaardfout wordt alleen gebruik gemaakt van de elementen op de hoofddiagonaal van de

informatiematrix.) Deze methode wordt automatisch gebruikt, indien het aantal te schatten parameters groter is dan 181.

item tSts. (Zie hoofdstuk 6.)

Name. Hier wordt de naam vastgelegd van alle bestanden die het programmapakket zal aanmaken. Deze naam moet een legale DOS-bestandsnaam zijn maar zonder een extentie (maximaal 8 tekens).

loG. (Yes/No) Aanmaken van het .LOG bestand.

paraM. (Yes/No) Aanmaken van het .PAR bestand.

coVar. (Yes/No) Aanmaken van het .COV bestand.

siZe. (Standard/Extended/Huge) (Zie ook hoofdstuk 6.) Hoeveelheid uitvoer (alleen voor OPCML).

taBles. (1/2) In de uitvoer worden de tabellen van de klassieke item-analyse en van de schattingen van de θ-waarden weggeschreven in één of twee kolommen. Eén kolom is aan te bevelen als men deze resultaten in andere programma's wil gebruiken; twee kolommen is overzichtelijker voor inspectie.

Title. Een willekeurige titel van maximaal 71 tekens. Deze titel verschijnt boven aan de uitvoer.

Komment. Hier kan men een willekeurig commentaar van maximaal 20 regels tekst invoeren. Dit commentaar verschijnt op de uitvoer onder de titel. Tijdens de invoer van het commentaar bevat het hulpscherm ^F1 hulp over het gebruik van de editor.

direQt. (Directory) De directory waar het gegevensbestand is te vinden. Wil men de gegevens inlezen van een floppy op station A, dan dient men hier op te geven A:. Staan de gegevens in de directory, dan mag men dit veld blank laten. (Algemeen: het programma OPPRE zoekt de gegevens in de directory die

hier is opgegeven; is er niets opgegeven, dan zoekt OPPRE in de directory die actief is op het moment dat OPPRE gestart wordt.)

fYle. De bestandsnaam van het gegevensbestand. Deze naam is willekeurig en heeft niets te maken met de naam die in het veld **Name** is opgegeven.

Format. Het formaat van de gegevens in het data-bestand. Een formaat geeft aan hoe er dient gelezen te worden. (In scherm 2 wordt precies aangegeven wat er gelezen moet worden.)
Om aan te geven dat er een geheel (integer) getal moet gelezen worden gebruikt men de letter I. Het aantal kolommen dat het getal inneemt wordt de veldbreedte genoemd. De veldbreedte wordt onmiddellijk achter de aanduiding 'I' geschreven. I1 betekent een getal dat één kolom beslaat; I3 betekent een getal dat drie kolommen beslaat. Om aan te geven dat drie getallen met veldbreedte 1 achter achter elkaar staan kan men opgeven: I1, I1, I1. Omdat drie keer na elkaar dezelfde aanduiding gegeven wordt kan dit veel eleganter geschreven worden als 3I1. Het getal dat voor de I staat wordt de repetitiefactor genoemd. Als de repetitiefactor 1 is hoeft hij niet geschreven te worden. I1 is hetzelfde als 1I1. De repetitiefactor moet kleiner zijn dan 256.

Voorbeelden:

7I4 : betekent 7 gehele getallen van 4 cijfers. De getallen dienen rechts uitgelijnd in het veld van 4 cijfers te staan. Blanco's worden gelezen als nul. Verder mag het getal alleen uit cijfers en eventueel een minus-teken bestaan.

I4,5I4,I4 betekent exact hetzelfde als 7I4.

300I1 wordt door de computer niet goed geïnterpreteerd. Wil men 300 getallen van 1 cijfer inlezen, dan kan dat door 200I1,100I1

De X geeft aan dat er kolommen moeten worden overgeslagen. X moet altijd een repetitiefactor hebben. Voorbeeld:

1X betekent 'sla één kolom over'.

T is een absolute 'tab'. T30 betekent 'ga op kolom 30 staan'.

Het is belangrijk te weten dat een Format 'te groot' mag zijn voor een record, maar niet 'te klein'. Moet het programma bijvoorbeeld van een record 10 ééncijferige getallen lezen die in de kolommen 1 to 10 staan, dan kan dit met het Format (10I1), maar ook met het Format (154I1). Het kan niet met het Format (9I1).

Voorbeelden

(1) Voor de gegevens afgebeeld in Figuur 6 kan het Format (I2,8X,10I1) worden gebruikt.

(2) Voor de gegevens in Figuur 7 kan eveneens het Format uit voorbeeld (1) gebruikt worden. Het volgende Format is echter ook goed: (I2,8X,7I1). Bemerk dat het Format (I2,8X,6I1) niet goed is, want het is niet voldoende om een record van boekje 1 in te lezen. Indien men toch dit Format zou gebruiken, krijgt men daar geen foutmelding over. De gegevens worden ingelezen met het formaat dat de gebruiker heeft opgegeven, en het programma kan niet weten dat de gebruiker iets anders bedoelt dan hij aangeeft. Meestal zal men echter wel eigenaardige resultaten verkrijgen. Het is dan ook van het allergrootste belang dat het Format goed gecontroleerd wordt.

(3) Het programma OPPRE voert de opdracht uit om bij elk gelezen record het eerste getal dat hij leest, te interpreteren als het boekjesnummer. Als het boekjesnummer achteraan staat moet men er middels het Format voor zorgen dat dit getal toch als eerste wordt ingelezen. Stel dat voor de gegevens in Figuur 6 het boekjesnummer niet in de kolommen 1 en 2 staat, maar in de kolommen 21 en 22. Het Format voor dit geval is (20X,I2,T11,7I1) hetgeen betekent: 'sla de eerste twintig kolommen over (20X), lees een getal van twee cijfers (I2), ga naar kolom 11 (T11) en lees 7 (of minder) getallen van 1 cijfer (7I1)'. Door het

gebruik van de T in het Format verloopt het lezen van de gegevens trager. Het verdient dus aanbeveling het boekjesnummer vóór de itemantwoorden te plaatsen.

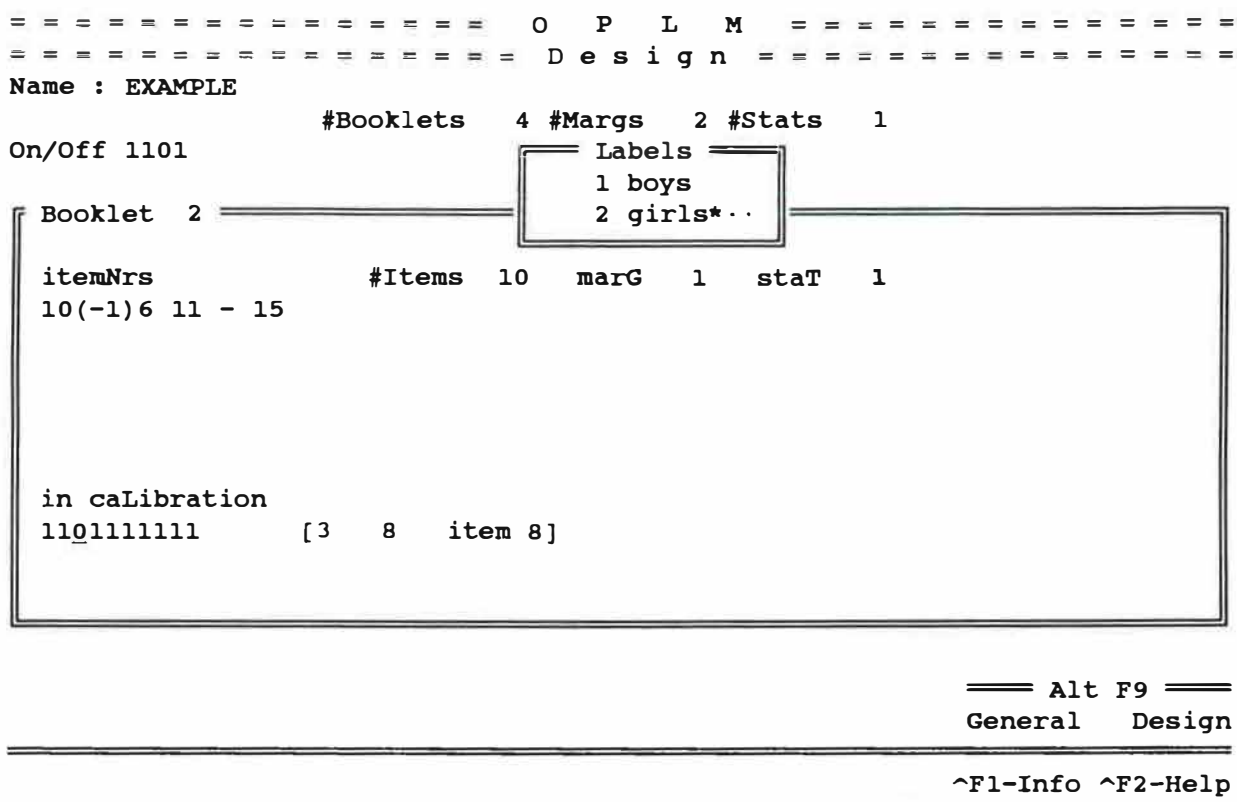
strUcture. (Expanded/Compressed) Hiermee geeft men de structuur van het gegevensbestand aan. (Zie paragraaf 3.1.)

3.3.3 Scherm 2: DESIGN

Figuur 12 is een weergave van scherm 2: DESIGN voor een fictief voorbeeld. Het teken '*' geeft aan waar de cursor staat. De toetscombinatie ^A laat toe heen en weer te springen tussen scherm 1 en 2. De velden Name en Booklet zijn niet bereikbaar met de cursor. Name geeft de naam aan die in scherm 1 is gekozen. Het kader (subscherm) waar het veld Booklet deel van uitmaakt, moet voor elk boekje gespecificeerd worden. In Figuur 12 is de specificatie aangegeven voor boekje 2.

#Booklets. (maximum 70) Het aantal boekjes dat in het gegevensbestand voorkomt. Strikt genomen wordt in dit veld het hoogste boekjesnummer opgenomen dat in de analyse zal betrokken worden. In Figuur 12 is dit 4. Indien er in het gegevensbestand boekjesnummers voorkomen die groter zijn dan dit getal, worden die records overgeslagen en geteld. Op de uitvoer worden ze vermeld als 'illegal records'.

On/Off. (1/0) Per boekje kan men aangeven of het meedoet in de analyse of niet. In het voorbeeld is aangegeven dat de records die behoren tot boekje 3 in de analyse niet meedoen. Deze records worden bij het inlezen van het gegevensbestand overgeslagen en geteld. Op de uitvoer worden ze vermeld als 'skipped records'. Dit veld en het veld #Booklets laten toe de boekjesnummers niet te laten starten bij 1. Stel dat er vier boekjes zijn die op het gegevensbestand zijn aangegeven met de getallen 5 6 7 en 8. Om ze allemaal in de analyse te laten meedoen geeft men 8 op als #Booklets en 00001111 als On/Off.



Figuur 12

Scherm 2 (DESIGN) voor een fictief voorbeeld

#Margs. Aantal marginale verdelingen. Indien #Margs = 1, geeft men aan dat de hele steekproef als een aselecte steekproef uit één normale verdeling is getrokken. Men kan echter ook veronderstellen dat er verschilende steekproeven zijn die elk uit een normale verdeling zijn getrokken. In Figuur 12 is aangegeven dat er twee populaties zijn. Bovendien kan men die populaties een naam geven (zie het subschermpje 'Labels'). Het programma OPMML zal dan naast de itemparameters twee gemiddelden en twee varianties schatten. Wel geldt de restrictie dat alle personen die hetzelfde boekje gekregen hebben uit dezelfde populatie komen (zie bij marG). Indien ook klassieke toets-en item analyse wordt gevraagd (zie het veld 'Classical' in scherm 'General') worden de p-waarden van de items uitgerekend per populatie. Deze berekening gebeurt door het programma OPPRE. De uitvoer van de klassieke analyse wordt weggeschreven in het .PRE-bestand, en kan zichtbaar gemaakt worden met het programma OPLOOK. De uitvoer van de klassieke analyse verschijnt ook in het .CML-bestand.

#Stats. (Zie hoofdstuk 6.)

Specificaties per boekje. De volgende velden moeten voor elk boekje worden opgegeven, ook voor boekjes die niet aan de analyse meedoen. Het boekjesnummer waarvoor men de specificaties opgeeft, staat links boven het subscherm. Men kan makkelijk de subschermen doorlopen door gebruik te maken van de toetsen F8 (vooruit) of F7 (achteruit).

#Items. (maximum 225) Het aantal items per boekje, inclusief de items die in scherm 1 zijn uitgezet in het veld cAl.

marG. Het populatienummer van het boekje. (Zie ook #Margs.)

staT. (Zie hoofdstuk 6.)

itemNrs. De (absolute) itemnummers van de items die in dit boekje zijn opgenomen. Deze nummers moeten overeenkomen met de nummers in het veld Nr van scherm 1. De volgorde waarin de nummers worden opgegeven, moet overeenkomen met de volgorde waarop ze in het gegevensbestand zijn opgenomen.

Intervallen kunnen opgegeven worden met een '-': 1-5 is equivalent met 1 2 3 4 5; 5-1 is equivalent 5 4 3 2 1. Intervallen of afzonderlijke getallen worden gescheiden door een komma of door één of meerdere blanco's. Bij rijtjes die met regelmatige stappen toenemen of afnemen, wordt de stapgrootte tussen haakjes aangegeven tussen de eerste en de laatste waarde. 1(2)5 is equivalent met 1 3 5; 10(-1)6 is equivalent met 10 9 8 7 6.

in caLibration. (1/0) Items kunnen lokaal uitgezet worden. Van items die in dit veld zijn uitgezet, doen de antwoorden niet mee, voor zover ze in het betreffende boekje voorkomen; wel als ze in andere boekjes voorkomen. Items die in scherm 1 in het veld cAl zijn uitgezet, worden niet opgenomen in de analyse; hun status in de subschermen van scherm 2 is dan irrelevant. Hetzelfde geldt voor boekjes die zijn uitgezet: als een boekje is uitgezet kunnen zijn items niet lokaal worden aangezet.
In Figuur 12 duidt de onderstreping de cursor aan, wanneer het veld 'in caLibration' actief is. Het getal boven de cursor wordt toegelicht tussen

[ ] rechts: het gaat om het 3e item in dit boekje; dit item is itemnummer 8 met het itemlabel 'item 8'.

3.3.4 Het SAVE-menu

Het Save-menu kan bereikt worden met F10 vanuit scherm 1 of 2. Figuur 13 geeft een afbeelding van het Save-menu na het invullen van de schermen waarbij de naam 'Example' werd gebruikt.

```
O P I N   Save-Menu
Directory   C:\ANALYSE
File        EXAMPLE.SCR
Start File
No Save
```

Figuur 13

Het SAVE-menu

Directory. Naam van de directory waar het .SCR-bestand weggeschreven moet worden. Indien dit veld niet wordt ingevuld, wordt het bestand weggeschreven in de directory die actief is.

File. Het programma vult zelf de naam van het .SCR-bestand in; het is de naam die door de gebruiker in het veld 'Name' van scherm 1 is ingevuld. Door [CR] geeft men aan dat men het bestand onder die naam ook wil bewaren. Indien een bestand van die naam reeds bestaat, geeft het programma een waarschuwing: men kan het bastaande bestand overschrijven, of een nieuwe naam aan het huidig bestand geven. Doet men dit laatste, dan wordt ook automatisch de naam in het veld 'Name' van scherm 1 in overeenstemming gebracht met de nieuw gekozen bestandsnaam.

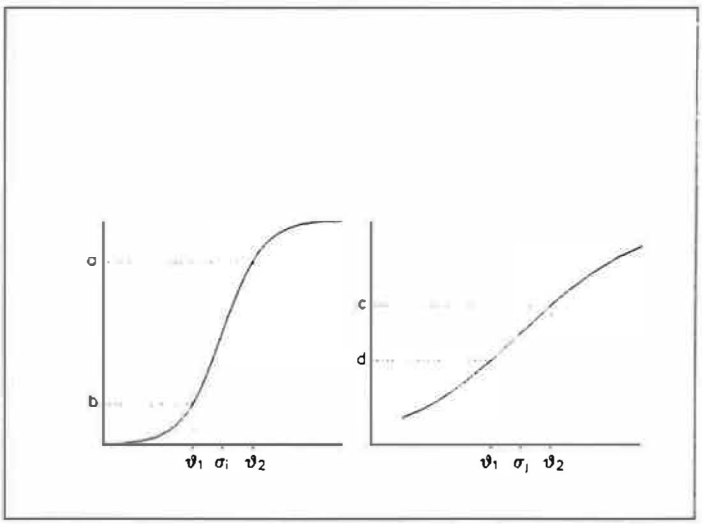
Start File. Men kan het bestand ook wegschrijven als Start File. Deze naam zal in het Init menu in het veld Start File verschijnen.

No Save. De aangebrachte veranderingen worden niet weggeschreven.
Na het verlaten van het Save-menu verschijnt het Init-menu (paragraaf 3.3.1) opnieuw op het scherm. Dit kan men verlaten met de keuze 'Quit'.

4 Het éénparameter logistisch model (OPLM)

4.1 De modelvergelijking

Om een indruk te krijgen van de strengheid van de assumpties die aan het Rasch-model ten grondslag liggen, beschouwen we Figuur 14.



Figuur 14

Twee items die verschillend discrimineren

Daarin zijn twee itemresponscurves afgebeeld voor de items i en j. We nemen aan dat $\sigma_i = \sigma_j$. Beschouw nu twee personen met latente vaardigheden ϑ_1 en ϑ_2. Uit de figuur is duidelijk dat

$$f_i(\vartheta_2) - f_i(\vartheta_1) = a - b > c - d = f_j(\vartheta_2) - f_j(\vartheta_1) \qquad (20)$$

Dit betekent dat we op grond van item i een beter onderscheid kunnen maken tussen die twee personen dan op grond van item j, hoewel beide items even moeilijk zijn. Anders gezegd: item i discrimineert beter dan item j. Dit betere onderscheidingsvermogen komt tot uiting in het steiler verloop van de itemresponscurve van item i. Bemerk overigens dat dit onderscheidingsvermogen een plaatselijke eigenschap is: twee personen met een verschillende vaardigheid die voor beide veel groter is dan de

moeilijkheidsgraad van het item, zullen beide bijna zeker het item oplossen en dus kan het item geen onderscheid maken tussen beider vaardigheid. Het onderscheidend vermogen van een item wordt dus afgemeten aan de snelheid waarmee de itemresponsfunctie verandert in de buurt van de moeilijkheidsparameter. Als we binnen de familie van logistische functies blijven, kunnen we dit verschil in discriminerend vermogen uitdrukken door een iets gecompliceerder functievorm te kiezen dan in het Rasch-model. De formule voor functies zoals weergegeven in Figuur 14 is:

$$\text{Prob}(X_i=1|\vartheta) = f_i(\vartheta) = \frac{e^{a_i(\vartheta-\sigma_i)}}{1+e^{a_i(\vartheta-\sigma_i)}}, \quad (a_i>0), \tag{21a}$$

waaruit direct volgt dat

$$\text{Prob}(X_i=0|\vartheta) = 1-f_i(\vartheta) = \frac{1}{1+e^{a_i(\vartheta-\sigma_i)}}. \tag{21b}$$

De grootheid a_i wordt discriminatie-index of discriminatieparameter genoemd. (Op het onderscheid komen we verder terug.) Vergelijking met formule (19) leert dat het hier gaat om slechts één verandering van de algemene formule van het Raschmodel: de parameter a_i draagt een index, waarmee wordt aangegeven dat de waarde van die parameter per item kan verschillen, terwijl in formule (19) α geen index draagt, wat wil zeggen dat de waarde van α niet per item kan verschillen. Het Rasch-model is dus gekenmerkt door gelijke discriminatie voor alle items, en dat is een strenge veronderstelling.

We sluiten deze paragraaf af met een klein functie-onderzoek van (21a). Als a_i zeer dicht bij nul ligt, zal de exponent van e in (21a) ook zeer dicht bij nul liggen en de teller dus zeer dicht bij 1. Dus

$$\lim_{a_i \to 0} f_i(\vartheta) = \frac{1}{2}. \tag{22a}$$

Als a_i zeer groot wordt, is het functieverloop iets gecompliceerder. Indien $\vartheta > \sigma_i$ wordt het product $a_i(\vartheta-\sigma_i)$ zeer groot, en zal de teller van (21a) dus

ook zeer groot worden. Indien echter $\vartheta < \sigma_i$, wordt het product $a_i(\vartheta-\sigma_i)$ zeer groot in absolute waarde maar negatief, zodat de teller van (21) zeer dicht bij 0 zal liggen. Dus

$$\lim_{a_i \to \infty} f_i(\vartheta) = \begin{cases} 1 \text{ indien } \vartheta > \sigma_i, \\ 0 \text{ indien } \vartheta < \sigma_i, \\ \text{onbepaald indien } \vartheta = \sigma_i. \end{cases} \qquad (22b)$$

De grafiek van dit limietgeval is weergegeven in Figuur 1. We zien dus dat bij een zeer groot discriminerend vermogen het item een (bijna) perfect onderscheid maakt tussen personen met een vaardigheid aan weerszijden van de moeilijkheidsparameter.

4.2 Toegestane transformaties

Zoals in de paragrafen 2.5 en 2.6 is uiteengezet, kunnen we de latente variabele (ϑ) en de parameters transformeren. We hebben daar gezien dat zo'n transformatie neerkomt op het kiezen van een nulpunt en een eenheid van de schaal. Bij transformaties dienen we erop te letten dat de exponent van e in (21a) niet verandert. Kiezen we nu de volgende transformatie:

$$\begin{aligned} a_i^* &= c\, a_i, \\ \vartheta^* &= (\vartheta + b)/c, \\ \sigma_i^* &= (\sigma_i + b)/c, \quad (c>0), \end{aligned} \qquad (23)$$

dan is meteen duidelijk dat

$$a_i^*(\vartheta^* - \sigma_i^*) = a_i(\vartheta - \sigma_i). \qquad (24)$$

Daaruit volgt direct dat de absolute waarde van a_i niets vertelt over het discriminerend vermogen van het item; we kunnen immers een willekeurige positieve constante c kiezen, waarmee we a_i kunnen vermenigvuldigen. De verhouding tussen de discriminatieparameters van twee items blijft echter ongewijzigd onder de toegelaten transformaties:

$$\frac{a_i^*}{a_j^*} = \frac{ca_i}{ca_j} = \frac{a_i}{a_j}. \qquad (25)$$

We kunnen dus voor een aantal items waarvan de discriminatieparameters gegeven zijn, steeds een zodanige transformatie kiezen dat de getransformeerde discriminatieparameters gehele getallen zijn. Stel bijvoorbeeld dat voor k = 3,
$a_1 = 2.5$, $a_2 = 1$ en $a_3 = 1.5$
en we kiezen c = 2, dan is
$a_1^* = 5$, $a_2^* = 2$ en $a_3^* = 3$.

4.3 De status van a_i

In paragraaf 2.3 hebben we gezien dat we in het Rasch-model een conditionele aannemelijkheidsfunctie konden opstellen die onafhankelijk is van ϑ. Dit konden we alleen doen door te conditioneren op de score, dat is het aantal juist beantwoorde items. Omdat CML-schattingen grote theoretische voordelen hebben, willen we ook graag zulke schattingen hebben in het meer algemene model (21). Probeert men hier een conditionele aannemelijkheidsfunctie op te stellen die onafhankelijk is van ϑ, dan kan men aantonen dat er nu moet worden geconditioneerd op de gewogen score, dit is de score die ontstaat door de a-waarden van alle juist beantwoorde items bij elkaar op te tellen:

$$s = \sum_i^k a_i x_i. \qquad (26)$$

En hier rijst een probleem: om een conditionele schattingsmethode op te stellen, dient men van iedere respondent de score te kennen, maar uit (26) blijkt duidelijk dat we de score alleen kunnen berekenen als we de a_i-waarden kennen. En die kennen we in het algemeen niet. Nu kan men bij dit probleem twee standpunten innemen.

Het eerste standpunt, dat meestal ingenomen wordt in de psychometrische literatuur, stelt dat de a_i's net als de σ_i's uit de data dienen te worden geschat. De a_i's zijn dus onbekende parameters en per item moeten er twee parameters geschat worden. Indien men dit standpunt inneemt, wordt (21a) de

modelvergelijking van het 'twee-parameter logistisch model' genoemd (of Birnbaum-model, naar zijn uitvinder.) Dit standpunt brengt echter mee dat CML uitgesloten is.

Men kan echter ook de waarde van de a_i's per hypothese invoeren (dat wil zeggen doen alsof men ze kent). In dit geval hoeven ze niet uit de data geschat te worden en bevat (21a) maar één parameter, namelijk σ_i. Vandaar de naam éénparameter logistisch model. Dit standpunt heeft het voordeel dat er nu wel CML-schattingen gevonden kunnen worden. Om duidelijk aan te geven dat a_i als een bekende grootheid behandeld wordt, krijgt hij de naam 'discriminatie-index'. Verder zullen we steeds de a_i's als bekende constanten behandelen. Hoewel we nu het voordeel hebben dat CML kan worden toegepast, moet er voor dit standpunt wel een prijs betaald worden. Daarop komen we terug in hoofstuk 5.

Als illustratie stellen we de formule op voor de conditionle probabiliteit van een antwoordpatroon in het éénparameter logistisch model. Veronderstel $k=4$, $a_1 = a_2 = a_3 = 1$ en $a_4 = 2$. Beschouw het antwoordpatroon (1 1 0 0). De score die bij dit antwoordpatroon hoort, is $a_1 + a_2 = 2$. We moeten dus de probabiliteit bepalen dat precies dit antwoordpatroon verkregen wordt, gegeven dat de score gelijk is aan 2. Nu zijn er precies vier antwoordpatronen die een score 2 opleveren:

(1 1 0 0),
(1 0 1 0),
(0 1 1 0),
(0 0 0 1).

De gezochte conditionele probabiliteit is dus

$$\text{Prob}(1100|s=2,\hat{\theta}) = \frac{\text{Prob}(1100|\hat{\theta})}{\text{Prob}(1100|\hat{\theta})+\text{Prob}(1010|\hat{\theta})+\text{Prob}(0110|\hat{\theta})+\text{Prob}(0001|\hat{\theta})}. \qquad (27)$$

Als gevolg van de lokale stochastische onafhankelijkheid kan elke probabiliteit in het rechterlid van (27) geschreven worden als een product van 4 uitdrukkingen zoals (21a) voor een juist antwoord of (21b) voor een

fout antwoord. Bemerk dat de noemers van (21a) en (21b) identiek zijn. De noemer van het product is dus voor elk antwoordpatroon dezelfde. Stellen we deze noemer voor door K. De teller van (27) kan dus geschreven worden als

$$\text{Prob}(1\,1\,0\,0|\vartheta) = \frac{e^{1(\vartheta-\sigma_1)} \times e^{1(\vartheta-\sigma_2)} \times 1 \times 1}{K}$$

$$= \frac{e^{2\vartheta} \times e^{-\sigma_1-\sigma_2}}{K}.$$

Soortgelijke uidrukkingen vinden we voor de probabiliteiten van de antwoordpatronen (1 0 1 0) en (0 1 1 0). Voor het antwoordpatroon (0 0 0 1) vinden we

$$\text{Prob}(0\,0\,0\,1|\vartheta) = \frac{1 \times 1 \times 1 \times e^{2(\vartheta-\sigma_4)}}{K}$$

$$= \frac{e^{2\vartheta} \times e^{-2\sigma_4}}{K}.$$

De teller en alle termen in de noemer van (27) hebben dus $e^{2\vartheta}/K$ als gemeenschappelijke factor. We krijgen dus als uiteindelijk resultaat

$$\text{Prob}(1\,1\,0\,0|s=2,\vartheta) = \frac{e^{-\sigma_1-\sigma_2}}{e^{-\sigma_1-\sigma_2} + e^{-\sigma_1-\sigma_3} + e^{-\sigma_2-\sigma_3} + e^{-2\sigma_4}}. \qquad (28)$$

en deze probabiliteit is uitsluitend een functie van de itemparameters.

Bij de toepassing van CML gaat de meeste berekeningstijd zitten in het uitrekenen van uitdrukkingen zoals (28).

Tot slot van dit hoofdstuk nog een opmerking over de beperkingen van CML met het éénparameter logistisch model. Stel dat een toets bestaat uit 10 items, en we stellen a_1 tot en met a_9 gelijk aan 2 en a_{10} gelijk aan 1. Dan zal duidelijk zijn dat in alle antwoordpatronen die een even score opleveren, item 10 fout is beantwoord en bij een oneven score is item 10 juist beantwoord. Dit betekent dus dat we uit elke mogelijke score met zekerheid kunnen afleiden wat het antwoord geweest is op item 10. In dit

geval kunnen we CML niet toepassen, omdat er geen unieke waarden van de itemparameters zijn die de aannemelijkheidsfunctie maximaliseren. Dit betekent dus dat we enigzins beperkt zijn in de keuze van de discriminatie-indices die we per hypothese kunnen invoeren. Er zijn nog meer (en soms heel complexe gevallen) waar geen schattingen bestaan. Sommige worden door het computerprogramma OPCML ontdekt en gemeld; andere worden niet ontdekt, doch men kan meestal aan de resultaten zien dat de oplossing eigenlijk niet bestaat. We komen hierop terug in de volgende paragraaf.

4.4 Een voorbeeld

In het voorbeeld illustreren we eerst het effect van schaaltransformaties op de parameterschattingen en hun standaardfout. We gaan vervolgens in op de betekenis van de standaardfout. Om bepaalde effecten duidelijk te laten zien maken we hier gebruik van artificiële gegevens. Artificiële gegevens zijn gegevens die niet in het veld verzameld zijn bij echte leerlingen, maar die door de computer aangemaakt worden. Door de computer te instrueren zich volgens het model te gedragen en hem te voeden met de echte parameterwaarden kunnen we nagaan hoe nauwkeurig de parameters kunnen worden teruggeschat uit de data. Daarbij weten we dat de afwijkingen tussen de echte en de geschatte parameters uitsluitend aan toeval zijn te wijten, en niet tot stand zijn gekomen, omdat bijvoorbeeld het model niet geldig is. Een dergelijke manier van werken wordt 'simulatie' genoemd en is nogal populair in de psychometrie.

Het aanmaken of genereren van artificiële data gaat als volgt. Veronderstel dat we 1000 artificiële leerlingen een antwoord willen laten geven op 6 items volgens het Rasch-model. Bovendien willen we dat de ϑ-waarden van de 1000 getrokken leerlingen een aselecte steekproef zijn uit een standaard-normale verdeling (dat wil zeggen een normale verdeling met gemiddelde nul en variantie één). Er bestaan computerprogramma's die zo'n steekproef kunnen trekken. In het voorbeeld hebben de items als parameterwaarde: -1, -0.5, 0., 0., 0.5 en 1.

Stap 1: trek uit de standaardnormale verdeling een ϑ-waarde. Neem als voorbeeld $\vartheta = 0.3$

Stap 2: bereken voor die ϑ-waarde de kans op een juist antwoord volgens het Rasch-model, door gebruik te maken van formule (4). Dit geeft als resultaat

$f_1(0.3) = 0.786;$
$f_2(0.3) = 0.690;$
$f_3(0.3) = f_4(0.3) = 0.574;$
$f_5(0.3) = 0.450;$
$f_6(0.3) = 0.332.$

Stap 3: Gooi zes vervalste munststukken op. Deze muntstukken hebben een kans om 'munt' te vallen die precies gelijk is aan de waarden die in stap 2 zijn uitgerekend. Elk muntstuk komt met een item overeen. Valt het 'munt' dan wordt het itemantwoord als juist beschouwd. Het opgooien van deze muntstukken wordt door de computer geïmiteerd via een speciaal programma. (Ga na dat deze procedure data genereert die geheel aan het Rasch-model voldoen.)

De stappen 1 tot 3 worden 1000 maal herhaald. De aldus gegenereerde data kunnen dan geanalyseerd worden met OPCML of OPMML. De resultaten van de analyse zijn weergegeven in Tabel 1.

Voor beide analyses met discriminatie-indices gelijk aan 1 zien we dat we niet de 'echte' waarden van de item-parameters terugkrijgen, maar ook dat we niet dezelfde schattingen krijgen, hoewel dezelfde data geanalyseerd zijn. Dat komt, omdat beide methodes niet dezelfde functie maximaliseren en we bijgevolg in het algemeen niet dezelfde schattingen zullen vinden. Indien de verdeling van de ϑ-waarden niet te veel afwijkt van de normale verdeling, zullen de schattingen van de itemparameters die beide methodes opleveren erg gelijkend zijn. De twee CML-oplossingen zijn, hoewel op het eerste gezicht erg verschillend, equivalent. In de tweede CML-oplossing zijn de discriminatie-indices verdubbeld, zodat de schattingen van de itemparameters automatisch gehalveerd zijn (zie formule 23).

Tabel 1

Analyse resultaten met OPMML en OPCML
voor artificiële data (N=1000)

nr	label	MML (a_1=1) σ	MML (a_1=1) SE(σ)	CML (a_1=1) σ	CML (a_1=1) SE(σ)	CML (a_1=2) σ	CML (a_1=2) SE(σ)
1	Item 1	-.900	.067	-.900	.068	-.450	.034
2	Item 2	-.562	.065	-.560	.065	-.280	.032
3	Item 3	.050	.064	.049	.063	.025	.031
4	Item 4	.045	.063	.045	.063	.022	.031
5	Item 5	.407	.064	.405	.064	.203	.032
6	Item 6	.959	.068	.960	.068	.480	.034

De kolommen SE (Standard Error of Standaardfout) zijn een aanduiding voor de nauwkeurigheid waarmee de parameters geschat zijn. Stel dat we de hierboven vermelde procedure van data genereren en analyseren een groot aantal keren uitvoeren (telkens met dezelfde steekproefgrootte) en voor een bepaald item een histogram maken van de schattingen van de moeilijkheidsparameter. De theorie van de grootste aannemelijkheids-schatters zegt dat dit histogram goed benaderd kan worden door een normale verdeling waarvan het gemiddelde de echte parameterwaarde is. De standaarddeviatie van die verdeling wordt de standaardfout genoemd. De grootheden die in de kolom SE staan, zijn schattingen van deze standaardfout. Hierbij dienen echter de volgende opmerkingen in acht genomen te worden:

(1) Deze schattingen zijn alleen goed als de steekproef niet te klein is. Bij kleine steekproeven (zeg rond de 100 observaties) zal de gerapporteerde schatting ernaar tenderen een onderschatting van de echte standaardfout te zijn.

(2) Het is niet alleen zo dat de schattingen van steekproef tot steekproef variëren, de schattingen van de verschillende item-parameters zijn ook afhankelijk van elkaar. Om een goede schatting te krijgen van de standaardfout dient deze afhankelijkheid meegenomen te worden in de berekening van de standaardfout. Dit gebeurt automatisch, indien als schattingsprocedure de optie 'Complete' is opgegeven in het veld 'Est

proc' van het scherm 'general'. Geeft men als optie 'Diagonal' op, dan rekent het programma alsof de schattingen van de verschillende itemparameters onafhankelijke grootheden zijn. Het effect hiervan is dat de schattingen van de standaardfouten meestal iets groter uitvallen dan met de procedure 'Complete'.

(3) Een standaard-deviatie wordt uitgedrukt in dezelfde eenheden als de variabele. Door de discriminatie-indices met twee te vermenigvuldigen wordt de eenheid van de schaal gehalveerd en derhalve wordt de standaardfout ook gehalveerd. De twee CML-analyses waarover in Tabel 1 is gerapporteerd, zijn volstrekt equivalent en de schattingen hebben dezelfde nauwkeurigheid.

(4) De belangrijkste variabele die invloed heeft op de nauwkeurigheid van de schattingen is de steekproefgrootte: hoe groter de steekproef, hoe groter de nauwkeurigheid. Hierbij is de belangrijkste regel dat de standaardfout proportioneel afneemt met de vierkantswortel uit de steekproefgrootte. Wil men de standaardfout (ongeveer) halveren, dan moet de steekproef verviervoudigd worden. In Tabel 2 staan de parameterschattingen en hun standaardfout berekend op de eerste 250 waarnemingen van de gegevens die gebruikt zijn om Tabel 1 op te stellen.

Tabel 2

Analyse resultaten met OPMML en OPCML
voor artificiële data (N=250)

nr	label	MML (a_i=1) σ	MML (a_i=1) SE(σ)	CML (a_i=1) σ	CML (a_i=1) SE(σ)	CML (a_i=2) σ	CML (a_i=2) SE(σ)
1	Item 1	-.951	.138	-.948	.137	-.474	.068
2	Item 2	-.691	.133	-.692	.133	-.346	.066
3	Item 3	-.135	.127	-.137	.128	-.068	.064
4	Item 4	.210	.127	.207	.128	.104	.064
5	Item 5	.441	.128	.438	.129	.217	.065
6	Item 6	1.126	.139	1.132	.141	.566	.070

Tenslotte nog een opmerking over gevallen waar CML-schatters niet bestaan. In de vorige paragraaf werd gezegd dat zulke gevallen soms door het programma ontdekt en gemeld worden. Deze melding kan verschillende vormen aannemen, afhankelijk van het moment waarop het defect ontdekt wordt. Indien men de optie 'Complete' gebruikt bij de schattingsmethode, kan het gebeuren dat men de melding krijgt 'Information matrix not positive definite'. Dit betekent dat er geen CML-schatters bestaan. Soms kan het gebeuren dat CML-schatters niet bestaan, doch dat het programma dit niet ontdekt. In zo'n geval nemen één of meer standaardfouten vaak onwaarschijnlijk grote waarden aan (groter dan 10). Soms zijn de standaardfouten zo groot dat ze niet meer kunnen worden afgedrukt op de daartoe voorziene plaats. Dan worden er sterretjes '*****' afgedrukt.

Gebruikt men de optie 'Diagonal', dan kan het gebeuren dat het programma schijnbaar 'aanvaardbare' schattingen vindt, terwijl ze in werkelijkheid niet bestaan. Dit is het geval indien men onvolledige designs gebruikt die niet gelinkt zijn (bijvoorbeeld een design met twee boekjes die geen gemeenschappelijke items hebben). Daarom verdient het aanbeveling waar mogelijk de optie 'Complete' te gebruiken.

De MML-schattingsmethode is veel minder gevoelig voor dit soort problemen. Hoewel de gevallen waar deze methode geen schattingen oplevert niet volledig bekend zijn, blijkt in de praktijk dat schattingen alleen dan niet bestaan, indien één of meerdere items door iedereen of door niemand juist zijn beantwoord. Zelfs met niet gelinkte boekjes kunnen MML-schattingen berekend worden, als men tenminste aanneemt dat de steekproeven uit dezelfde populatie komen. Als er bijvoorbeeld twee boekjes zijn zonder gemeenschappelijke items, dan kan men MML-schattingen vinden als men 1 opgeeft voor '#Margs'. De gemeenschappelijke distributie van ϑ-waarden zorgt dan voor de noodzakelijke link.

5 Toetsen van het model

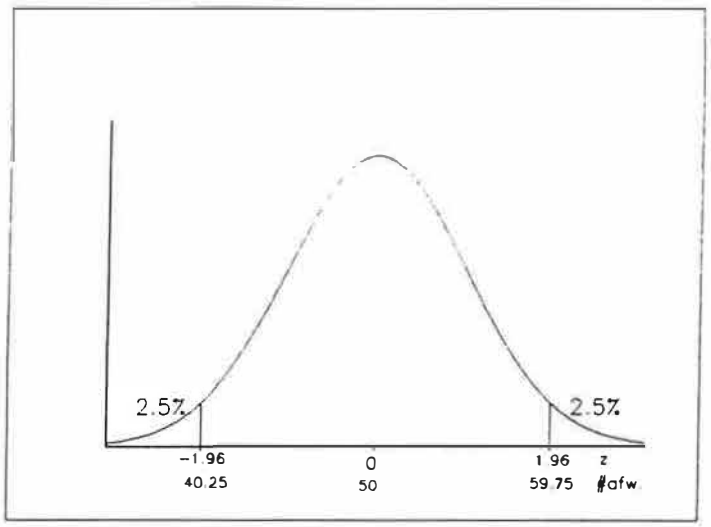
5.1 Rationale van de statistische toets

In hoofdstuk 2, paragraaf 2.1, hebben we de grootste-aannemelijkheids-schatter voor de parameter van een muntstuk opgesteld. Daar bleek dat we enkel het relatief aantal successen hoefden te kennen om die parameter te schatten. De observaties kunnen ons daarvoor niet meer informatie opleveren. Indien we zeker wisten dat aan de vooronderstellingen van het model was voldaan, hoefden we ook niet meer te weten. Bij het opgooien van een muntstuk zijn die vooronderstellingen eenvoudig: de kans op succes moet onveranderd blijven en de uitkomst bij elke worp moet onafhankelijk zijn van de andere worpen. Veronderstel nu dat het opgooien zo klungelig gebeurt dat de uitkomst bij een bepaalde worp bijna steeds gelijk is aan de uitkomst van de vorige worp, bijvoorbeeld omdat het muntstuk maar een klein beetje opgetild wordt en dan weer losgelaten. Als die afhankelijkheid heel sterk is, is het goed mogelijk dat bij 100 keer opgooien het muntstuk 99 keer 'munt' valt, ook al is het niet vervalst. We kunnen dan nog wel de techniek van het schatten toepassen, maar onze conclusie dat het muntstuk vervalst is, is niet terecht, omdat niet voldaan is aan de vooronderstellingen van het experiment.

We beschikken over mogelijkheden om te toetsen of aan die voor-onderstellingen is voldaan, door te kijken naar de fijnere structuur van de data, dat wil zeggen naar aspecten die we niet gebruikt hebben om de parameter te schatten. Indien we 100 keer een muntstuk opgooien en we stellen vast dat het muntstuk de eerste 50 keer 'munt' valt en de volgende 50 keer 'kruis' dan zullen we het muntstuk en/of het uitgevoerde experiment niet vertrouwen. We verwachten dat de afwisseling M-K of K-M meer dan één keer optreedt. Observeren we echter een volmaakte regelmaat waarbij K en M elkaar iedere keer weer afwisselen, dan is dit ook verdacht. Bemerk dat we in de redenering 'subjectieve' termen als 'vertrouwen' en 'verdacht' gebruiken en geen 'objectieve' termen als 'zeker' of 'onmogelijk'. Immers, ook als het experiment goed is uitgevoerd en er helemaal niets aan de hand is met het muntstuk is het theoretisch mogelijk dat het muntstuk de eerste 50 keer 'munt' valt en de volgende 50 keer 'kruis'. Als we in zo'n geval

het besluit nemen het experiment (of de veronderstellingen) niet te accepteren, maken we een fout. De kans op zo'n fout is in de statistiek niet uit te sluiten; we kunnen echter wel berekenen hoe groot die kans is.

Stel dat we 100 keer met een zuiver muntstuk op gooien en het aantal afwisselingen K-M of M-K tellen. Het minimum is 1 en het maximum is 99. Alle tussenliggende waarden kunnen voorkomen. In de statistiek is aangetoond dat dit aantal afwisselingen (#afw.) ongeveer normaal verdeeld is met een gemiddelde van 50 en een standaardafwijking van 4.975. De kans dat #afw. groter is dan 50 + 1.96*4.975 = 59.75 bedraagt 0.025 of 2.5%; de kans dat #afw. kleiner is dan 50 - 1.96*4.975 = 40.25 bedraagt eveneens 2.5% (zie Figuur 15). Ruwweg kunnen we dus stellen dat - bij een goed uitgevoerd experiment - het aantal afwisselingen tussen de 40 en de 60 zal liggen in 95% van de gevallen. Indien het geobserveerd aantal kleiner is dan 40 of groter dan 60, besluiten we dat, gezien de kans op zo'n extreme uitkomst kleiner of gelijk is aan 5%, de vooronderstellingen onaanvaardbaar zijn. Door deze regel consequent toe te passen weten we dat we in 5% van de gevallen het verkeerde besluit zullen nemen, indien het experiment toch goed is uitgevoerd. Die keuze van 5% is eigenlijk arbitrair. Die 5% draagt de algemene naam 'significantie niveau'. Vinden we een foutenrisico van 5% te hoog, en willen we bijvoorbeeld maar 1%, dan verwerpen we de vooronderstellingen indien het aantal afwisselingen kleiner is dan 50 - 2.58*4.975 = 37.16 of groter dan 50 + 2.58*4.975 = 62.84. Men zou kunnen redeneren dat het beter is het foutenrisico zo laag mogelijk te nemen (bijvoorbeeld 1 op 100000) en daarmee op safe te spelen. Zo eenvoudig is het echter niet. De fout die we in het voorgaande voorbeeld kunnen maken, bestaat erin dat we de veronderstellingen (het model) ten onrechte verwerpen. In de statistiek wordt dit een fout van de eerste soort genoemd. We kunnen echter ook een ander soort fout maken, namelijk de veronderstellingen niet verwerpen als ze fout zijn. Deze fout noemt men een fout van de tweede soort. Zie Tabel 3.
In het voorbeeld van het muntstuk zou er door de experimentele procedure een geringe afhankelijkheid in de reeks van uitkomsten kunnen sluipen waardoor het aantal afwisselingen bijvoorbeeld gemiddeld 60 is en niet 50.



Figuur 15

Steekproevenverdeling van het aantal afwisselingen bij het muntexperiment

Tabel 3

Beslissingstabel bij statistische toetsen

		werkelijkheid	
		veronderstelling waar	veronderstelling niet waar
beslissing	accepteren van veronderstelling	juiste beslissing kans = 1-α	fout van de tweede soort kans = β
	verwerpen van veronderstelling	fout van de eerste soort kans = α 'significantieniveau'	juiste beslissing kans = 1-β 'power'

Als we even aannemen dat in werkelijkheid het aantal afwisselingen normaal verdeeld is met een gemiddelde van 60 en een standaardafwijking van 4.975, dan ontstaat een situatie zoals afgebeeld in Figuur 16. De kans dat we een uitkomst krijgen die groter is 59.75, is de kans dat we onder de standaard normale verdeling een z-waarde krijgen groter dan (59.75-60)/4.975 = -0.05, en deze kans is (zie tabellen van de standaard normale verdeling) gelijk aan 0.52. Er is natuurlijk ook een positieve kans dat we een aantal vinden kleiner dan 40.25, doch deze kans is verwaarloosbaar klein. Dus als in werkelijkheid het aantal afwisselingen gemiddeld niet 50 is maar 60 en onze

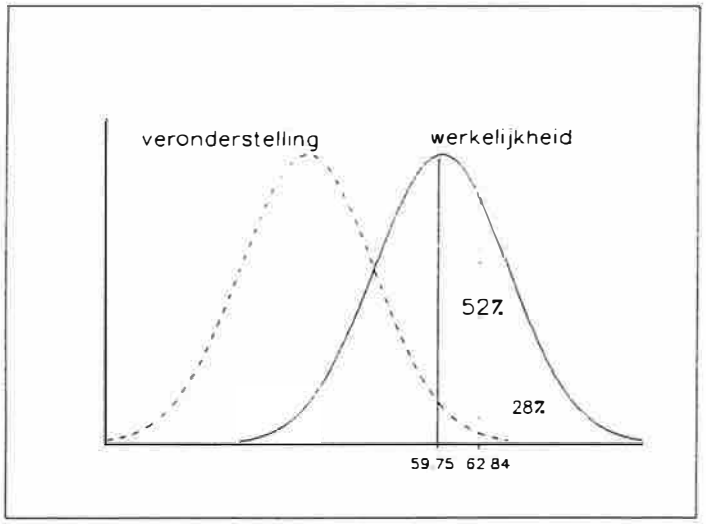
veronderstelling dus onwaar, is de kans dat we het juiste besluit nemen 52%. Deze kans noemt men het onderscheidingsvermogen (power) van de toets. De kans op het verkeerde besluit (β) is dus 1-0.52=0.48. Volgen we echter een procedure met een significantieniveau van 1%, dan zullen we de veronderstellingen pas verwerpen, indien we een uitkomst vinden groter dan 62.84. Deze kans is Prob[z > (62.84-60)/4.975)] = Prob(z > 0.57) = 0.28. We zien dus dat een vermindering van de kans op een fout van de eerste soort van 5% naar 1% gepaard gaat met een toename van de kans op een fout van de tweede soort van 48% naar (100-28)=72%. Dit is algemeen zo: verlaging van het significantieniveau betekent toename van de kans op een fout van de tweede soort, en dus vermindering van het onderscheidend vermogen van de toets.

Uit het bovenstaande moet duidelijk zijn dat α door de onderzoeker vastgelegd wordt en één welbepaalde waarde heeft: immers α geeft de kans op een fout aan indien de werkelijkheid zich gedraagt volgens de nauwkeurig door ons omschreven veronderstellingen. Het 'niet-waar-zijn' van de veronderstellingen kan echter veel dingen betekenen. Indien het gemiddeld aantal afwisselingen in werkelijkheid niet precies 50 is maar 50.01 zijn de veronderstellingen niet waar, maar het zal duidelijk zijn dat in zo'n geval het onderscheidend vermogen van de toets veel lager is dan in het geval van een gemiddelde van 60. (Maak een schetsje waaruit blijkt dat bij een α van 5%, het onderscheidend vermogen eveneens ongeveer 5% zal zijn: de twee verdelingen van het aantal afwisselingen, de veronderstelde en de werkelijke, vallen nagenoeg samen.) Het onderscheidend vermogen van een toets ($1-\beta$) is dus afhankelijk van de mate waarin de veronderstellingen niet waar zijn. In het algemeen geldt dat een statistische toets beter grove schendingen ontdekt dan kleine. ('Ontdekken' betekent 'leiden tot verwerping van de veronderstellingen als deze inderdaad niet waar zijn'.)

Het manipuleren van het onderscheidend vermogen van een toets

We hebben gezien dat het onderscheidend vermogen verandert door een ander significantieniveau te kiezen. Deze manier om het onderscheidend vermogen te reguleren is echter niet zo handig, omdat men meestal het significantieniveau op een bepaalde waarde wil instellen. De belangrijkste manier om het onderscheidend vermogen te reguleren is de keuze van de steekproefgrootte.

We illustreren dit aan de hand van een voorbeeld. We willen weten of jongens en meisjes gemiddeld verschillend presteren op een bepaalde toets. Stel dat in werkelijkheid de meisjes 2 punten hoger scoren dan de jongens en dat in de populatie van de jongens en van de meisjes de standaarddeviatie (σ) van de scores gelijk is aan 8.



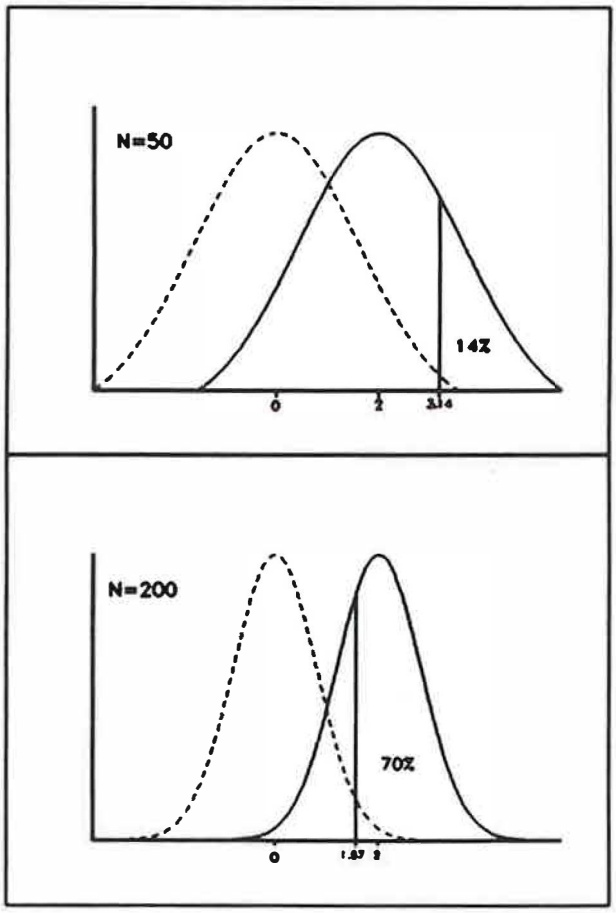
Figuur 16

Het onderscheidend vermogen (power) van de toets

Statistisch toetsen betekent dat er een veronderstelling wordt ingevoerd en dat gekeken wordt of de resultaten van de steekproef er niet 'verdacht' uitzien, als die veronderstelling waar is. De veronderstelling duidt men meestal aan met de naam 'nulhypothese'. De simpelste nulhypothese in dit voorbeeld luidt: $\mu_m - \mu_j = 0$, het populatiegemiddelde van de jongens en de meisjes is hetzelfde. Als de nulhypothese waar is, dan verwachten we eigenlijk dat het verschil tussen de steekproefgemiddelden ($D = M_m - M_j$) van de jongens en de meisjes niet al te groot is. Dit verschil is normaal verdeeld met gemiddelde $\mu_m - \mu_j$ en standaarddeviatie

$$\sigma_D = \sqrt{\frac{2\,\sigma^2}{N}}$$

waarbij N de grootte van de steekproef is, dat wil zeggen er zijn N jongens en N meisjes geobserveerd. Deze standaarddeviatie wordt de standaardfout van het gemiddeld verschil genoemd en uit de formule volgt heel duidelijk dat deze standaardfout kleiner wordt als N toeneemt. Onderzoeker 1 wil de nulhypothese toetsen met N=50 en onderzoeker 2 met N=200. Bij onderzoeker 1 is de standaardfout van het gemiddeld verschil $\sqrt{(2*8^2)}/50 = 1.6$ en bij onderzoeker 2 $\sqrt{(2*8^2)}/200 = 0.8$. Als ze beide op het 5% niveau toetsen zal onderzoeker 1 de nulhypothese verwerpen, als het gevonden verschil tussen de twee steekproefgemiddelden in absolute waarde groter is dan $1.96*1.6 = 3.14$ en onderzoeker twee bij een verschil groter dan $1.96*0.8 = 1.57$. De kans dat beide onderzoekers de nulhypothese (terecht) zullen verwerpen bij een werkelijk verschil van $\mu_m - \mu_j = 2$ is duidelijk af te lezen uit Figuur 17.



Figuur 17

Onderscheidend vermogen als
functie van de steekproefgrootte

Onderzoeker 1 heeft slechts 14% kans om de nulhypothese te verwerpen, terwijl onderzoeker 2 een kans heeft van 70%. Dit voorbeeld maakt ook duidelijk dat fouten van de eerste en van de tweede soort niet gewoon als broertjes moeten worden opgevat: de kans op een fout van de eerste soort ligt vast en is onafhankelijk van de steekproefgrootte. Het ten onrechte verwerpen van de nulhypothese hebben we volledig in de hand. De kans op een fout van de tweede soort hebben we maar ten dele in de hand, door het manipuleren van de steekproefgrootte. Het onderscheidend vermogen is echter ook afhankelijk van de mate waarin de nulhypothese geschonden is. Nemen we bij een kleine tot matige schending een te kleine steekproef, dan zullen we weinig onderscheidend vermogen hebben. Als we geen significantie vinden, rechtvaardigt dat echter geenszins de uitspraak dat 'bewezen is dat er geen verschil is'. We hebben alleen niet kunnen aantonen dat er verschil bestaat. Omgekeerd echter kunnen we de steekproefgrootte zo groot maken dat we een triviaal klein verschil met grote kans ontdekken. Het vinden van een significant verschil betekent dus niet automatisch dat dit verschil belangrijk is. Het gedetailleerd onderzoek naar het onderscheidend vermogen van een toets vergt dus niet alleen statistisch inzicht, maar vereist ook een goed inhoudelijk inzicht in wat belangrijke en wat triviale verschillen zijn. Zo'n onderzoek is in de regel behoorlijk ingewikkeld en hier kan er niet verder op ingegaan worden.

In het vorige voorbeeld hebben we de veronderstellingen aangeduid met de term 'nulhypothese'. In de gewone toepassingen van de statistische theorie (afkomstig uit de experimentele landbouwkunde) hoopt men in de regel te nulhypothese te kunnen verwerpen: de nulhypothese verwoordt als het ware een pessimistisch standpunt ('bemesting heeft geen effect'), en het verwerpen van de nulhypothese geldt dan als bewijs van het effect van de experimentele behandeling. Bij de statistische toetsing van mathematische modellen neemt men het omgekeerde standpunt in: het model zelf heeft de status van nulhypothese, en men hoopt de nulhypothese niet te hoeven verwerpen. Men kan daarbij gemakkelijk zichzelf om de tuin leiden door zeer kleine steekproeven te hanteren, zodat de toets weinig onderscheidend vermogen heeft. Bovendien is het onderzoek naar het onderscheidend vermogen bij mathematische modellen extra moeilijk, omdat de modellen doorgaans complex zijn. In het vorige voorbeeld was dat onderzoek eigenlijk zeer

eenvoudig: de schending van de nulhypothese (geen verschil in prestatie tussen jongens en meisjes) kan maar één ding betekenen: het verschil is niet gelijk aan nul. Mathematische modellen zijn echter zeer complexe nulhypothesen en een schending kan van alles betekenen. Een schending van het OPLM-model zou kunnen betekenen: (1) misspecificatie van één of meerdere discriminatie-indices; (2) bias in één of meerdere items; (3) de itemresponscurve is niet monotoon voor één of meerdere items; (4) de items doen niet allemaal een beroep op dezelfde vaardigheid (multidimensionaliteit); (5) de veronderstelling van lokale onafhankelijkheid is geschonden; (6) er wordt geraden bij één of meerdere items. Bovendien kunnen meerdere schendingen tegelijkertijd aanwezig zijn, en het is erg moeilijk om deze schendingen te kwantificeren. De analyse naar het onderscheidend vermogen bij dergelijke modellen staat nog in de kinderschoenen en resultaten van de modeltoetsen moeten omzichtig worden behandeld.

5.2 Itemgerichte toetsen in OPLM

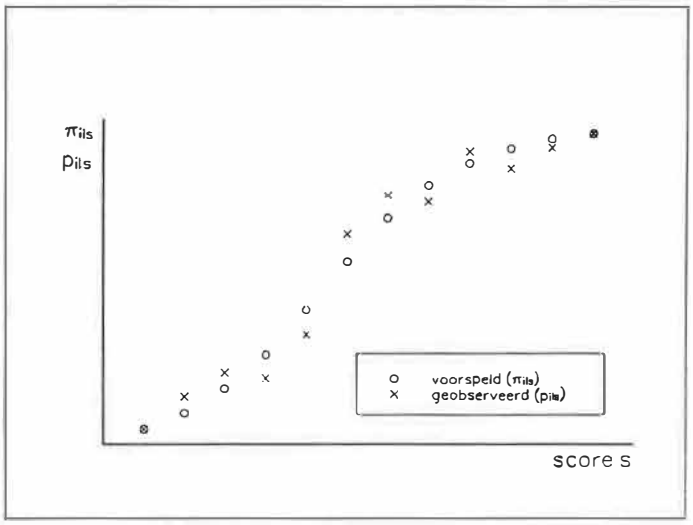
5.2.1 De X^2-toetsen

Om de itemparameters te schatten hebben we in OPLM maar twee soorten informatie nodig: van elke persoon moeten we de score kennen en van elk item dienen we te weten hoe vaak het juist is beantwoord. (In een volledig design komt dit overeen met de p-waarde.) Beschouw nu een willekeurig item i. Omdat een hoge score een aanduiding is van een grote ϑ-waarde, verwachten we dat de kans op een juist antwoord op item i groter is bij een hoge score dan bij een lage score. Als de itemparameters bekend zijn, kan deze kans precies worden bepaald. (De formule daarvoor wordt hier niet afgeleid.) Deze kans is dus een functie van de itemparameters en die kans zullen we aanduiden als $\pi_{i|s}$. Deze kans moet dus gelezen worden als

$$\pi_{i|s} = \text{Prob}(X_i=1 \,|\, \text{score}=s).$$

Indien we over schattingen van de itemparameters beschikken, kunnen we deze kans dus ook uitrekenen met de geschatte itemparameters. Bovendien kunnen

we in elke groep personen met score s bepalen welke proportie personen een juist antwoord heeft gegeven op item i. Deze proportie duiden we aan als $p_{i|s}$. Als het model juist is, dan moet deze proportie ongeveer gelijk zijn aan de probabiliteit $\pi_{i|s}$ (zie Figuur 18).
Volledige gelijkheid kunnen we niet eisen, omdat het aantal personen met score s in de steekproef eindig is. (Net zomin als we mogen eisen dat bij 100 keer opgooien een onvervalst muntstuk precies 50 keer 'munt' valt.) De vraag is dus of de afwijkingen tussen $\pi_{i|s}$ en $p_{i|s}$ aan het toeval kunnen worden toegeschreven (= nulhypothese), dan wel of er iets anders aan de hand is, namelijk dat de vooronderstellingen (het model) niet geldig zijn. Let wel: het is absoluut onmogelijk om uitsluitend aan de hand van Figuur 18 te beoordelen of de afwijkingen aan toeval zijn te wijten. (Hetzelfde geldt voor het experiment met het muntstuk: als je weet dat een muntstuk in 60% van de gevallen 'munt' is gevallen, kun je geen zinnige uitspraken doen over het al dan niet vervalst zijn van dit muntstuk.)



Figuur 18

Geobserveerde en voorspelde proporties juiste antwoorden als functie van de score s

In Tabel 4 zijn de voorspelde en geobserveerde proporties voor de scores 1 tot en met 12 weergegeven. Deze gegevens hebben betrekking op een toets van 13 items die met het Rasch-model zijn geanalyseerd. De minimale score is

nul en de maximale score 13. Deze twee scores zijn niet in de tabel opgenomen, omdat $\pi_{i|0} = p_{i|0} = 0$ en $\pi_{i|13} = p_{i|13} = 1$. In de kolom n_s is het aantal personen vermeld dat score s heeft behaald.

Tabel 4

Geobserveerde en verwachte proporties juiste antwoorden en aantal observaties per scoregroep

score	$p_{i\|s}$	$\pi_{i\|s}$	n_s
1	0.040	0.047	25
2	0.071	0.081	98
3	0.150	0.124	133
4	0.133	0.182	150
5	0.260	0.317	169
6	0.541	0.499	207
7	0.691	0.642	194
8	0.673	0.702	107
9	0.816	0.773	98
10	0.771	0.837	70
11	0.890	0.891	73
12	0.920	0.924	50

Een voor de hand liggende manier om de nulhypothese te toetsen is het omvormen van Tabel 4 tot een 12 maal 2 contingentietabel en daarop de 'gewone chi-kwadraat' toets toe te passen. Deze tabel is weergegeven in Tabel 5.
De geobserveerde frequenties O_{s1} (het aantal personen met score s die het item juist hebben) worden berekend met de formule $O_{s1} = n_s p_{i|s}$ en natuurlijk moet gelden $O_{s0} = n_s - O_{s1}$. De verwachte frequenties E_{s1} en E_{s0} worden op analoge manier berekend en staan tussen haakjes in de tabel. De bekende Pearson's chi-kwadraat toetsingsgrootheid wordt uitgerekend met de formule

$$X^2 = \sum_{s=1}^{12} \sum_{j=0}^{1} \frac{(O_{sj} - E_{sj})^2}{E_{sj}}. \qquad (29)$$

Toepassing van formule (29) op Tabel 5 levert op: $X^2 = 13.058$. Om te beoordelen of deze waarde 'verdacht' groot is, moeten we weten wat we

doorgaans kunnen verwachten bij een soortgelijke tabel, indien onze veronderstellingen waar zijn. De theoretische statistiek leert ons dat de

Tabel 5

Geobserveerde en (verwachte) frequenties voor juiste en foute antwoorden

score	juist antwoord	fout antwoord
1	1 (1.17)	24 (23.83)
2	7 (7.94)	91 (90.06)
3	20 (16.49)	113 (116.51)
4	20 (27.30)	130 (122.70)
5	44 (53.57)	125 (115.43)
6	112 (103.29)	95 (103.71)
7	134 (124.55)	60 (69.45)
8	72 (75.11)	35 (31.89)
9	80 (75.75)	18 (22.25)
10	54 (58.59)	16 (11.41)
11	65 (65.04)	8 (7.96)
12	46 (46.20)	4 (3.80)

grootheid X^2 die we gevonden hebben, afkomstig is uit een familie vanverdelingen die bekend staat onder de naam 'chi-kwadraat'. Het speciale lid van die familie dat op ons voorbeeld van toepassing is, heeft te maken met het aantal cellen in de tabel (en dus met het aantal termen in de formule (29)). Daarop moet echter een correctie worden toegepast, omdat

niet alle cellen vrij kunnen variëren: de som van de elementen in rij s is noodzakelijkerwijze gelijk aan n_s. Als we één element in een rij kennen, kennen we automatisch het andere, want n_s is gegeven. Bovendien is er nog een andere afhankelijkheid: de som van de geobserveerde frequenties in kolom 1 is gelijk aan de som van de verwachte frequenties. Deze gelijkheid hebben we namelijk geëist om de parameters te schatten. Er zijn in de tabel dus maar 11 vrije cellen, en niet 12x2=24. Het aantal vrije cellen wordt het aantal vrijheidsgraden genoemd (in het Engels: degrees of freedom). De gevonden grootheid van 13.058 moeten we dan ook vergelijken met de theoretische chi-kwadraatverdeling met 11 vrijheidsgraden. In tabellen over de chi-kwadraatverdeling (in elk statistiekboek te vinden) vinden we dat de kritieke waarde (op het 5% niveau) in een chi-kwadraatverdeling met 11 vrijheidsgraden gelijk is aan 19.675. Dit betekent dat, indien het model waar is, de gevonden X^2-waarde in 95% van de gevallen kleiner of gelijk aan 19.675 zal zijn. Wij hebben een waarde gevonden van 13.058, die dus behoort tot de 95% 'onverdachte' uitkomsten. We hebben dus geen reden om op basis van deze toets het model te verwerpen. (De exacte overschrijdingskans van de gevonden waarde is iets groter dan 25%: dat wil zeggen dat, indien het model waar is, in iets meer dan 25% van de gevallen een X^2-waarde zal gevonden worden die groter is dan 13.058. De exacte overschrijdingskans is niet gemakkelijk te berekenen. In het programma OPLM wordt ze wel berekend.)

Gebruik makend van de regel $O_{s1} = n_s p_{i|s}$ en $E_{s1} = n_s \pi_{i|s}$ kunnen we (29) herschrijven als

$$
\begin{aligned}
X^2 &= \sum_{s=1}^{12} \frac{(O_{s1} - E_{s1})^2}{E_{s1}} + \sum_{s=1}^{12} \frac{(O_{s0} - E_{s0})^2}{E_{s0}} \\
&= \sum_{s=1}^{12} \frac{n_s^2 (p_{i|s} - \pi_{i|s})^2}{n_s \pi_{i|s}} + \sum_{s=1}^{12} \frac{n_s^2 (p_{i|s} - \pi_{i|s})^2}{n_s (1 - \pi_{i|s})} \qquad (30) \\
&= \sum_{s=1}^{12} \frac{n_s^2 (p_{i|s} - \pi_{i|s})^2}{n_s \pi_{i|s} (1 - \pi_{i|s})}
\end{aligned}
$$

Hoewel het eindresultaat in (30) nog verder kan vereenvoudigd worden door teller en noemer te delen door n_s, laten we de uitdrukking even staan, omdat er nog modificaties moeten worden aangebracht. Er zijn immers een

paar overwegingen waardoor de bovenstaande redenering strikt genomen niet juist is.

1 De verwachte frequenties zijn een functie van de itemparameters; doch die itemparameters kennen we niet; we beschikken alleen over schattingen. De noodzakelijke correctie is mathematisch nogal ingewikkeld en wordt hier niet in detail besproken. De correctie wordt in het programma OPLM toegepast op aanvraag van de gebruiker: door in het veld 'item tSts' de optie 'Complete' te kiezen wordt de correctie toegepast. Kiest men 'Diagonal', dan wordt de correctie achterwege gelaten en is slechts de berekende toetsingsgrootheid een benaderende waarde. (Voor meer details, zie hoofdstuk 6.) Hoe goed die benadering in het algemeen is, is moeilijk te zeggen. Indien men echter beide methodes toepast, zal men merken dat in veel gevallen beide uitkomsten weinig van elkaar verschillen. Bovendien kan de correctie vanwege numerieke problemen niet worden toegepast, als het aantal te schatten parameters groter is dan 181; in zo'n geval gaat het programma automatisch over op de benaderende methode.

2 Een tweede overweging is dat de grootheid die met formule (29) of (30) berekend wordt, slechts de chi-kwadraatverdeling volgt als de steekproef 'oneindig' groot is. Steekproeven in de praktijk zijn natuurlijk altijd eindig, maar het gebruik van de chi-kwadraatverdeling is gerechtvaardigd, indien de verwachte frequenties in de tabel niet al te klein zijn. Bij realistische toepassingen op het Cito met gebruik van OPLM heeft men natuurlijk in de regel meer dan 13 items. Stel dat men een toets analyseert met 100 items en de discriminatiewaarden gemiddeld gelijk zijn aan 3, dan zijn er 300 verschillende scores mogelijk. Bij een steekproef van 3000 leerlingen is de gemiddelde score frequentie dus gelijk aan 10 en er zullen dus scores zijn die in de steekproef voorkomen met een zeer lage frequentie. Stel dat er precies 2 leerlingen zijn die score 294 hebben gehaald, dus $n_{294} = 2$. Omdat $E_{s0} + E_{s1} = n_s$, zijn beide verwachte frequenties klein. Men kan dus verwachten dat de frequentietabel (analoog aan Tabel 5) nogal wat kleine verwachte waarden zal bevatten. In zo'n geval volgt de grootheid die men met formule (29) uitrekent, niet meer de theoretische chi-kwadraatverdeling indien het model juist is. In zo'n geval groepeert men verschillende scores in

enkele groepen G_q, q = 1, ..., r. In het voorbeeld kiezen we r = 4, en $G_1 = \{1,2,3\}$, $G_2 = \{4,5,6\}$, $G_3 = \{7,8\}$ en $G_4 = \{9,10,11,12\}$. Als geobserveerde frequentie in zo'n groep neemt men de som van de geobserveerde frequenties van de samenstellende scores en idem dito voor de verwachte frequenties. De formule die men dan gebruikt is

$$X^2 = \sum_{q=1}^{r} \frac{\left[\sum_{s \in G_q} n_s (p_{i|s} - \pi_{i|s})\right]^2}{\sum_{s \in G_q} n_s \pi_{i|s} (1 - \pi_{i|s})} \qquad (31)$$

(Ga na dat, indien elke groep slechts uit één enkele score bestaat, formule (31) en (30) aan elkaar gelijk zijn.) Voor een goed begrip van formule (31) is het het belangrijkste de teller te inspecteren. (We gaan hier niet verder in op de technische details van de precieze structuur van de noemer. In het algemeen wordt de noemer zo geconstrueerd dat de resulterende grootheid X^2 (ongeveer) chi-kwadraat verdeeld is.) In de teller van (30) wordt voor elke score het verschil tussen geobserveerde en verwachte frequentie gekwadrateerd. Een verschil van -2 levert dezelfde bijdrage aan de teller als een verschil van +2: $(-2)^2 = 2^2 = 4$. X^2 kan dus zijn minimale waarde van nul slechts aannemen, indien alle geobserveerde frequenties precies gelijk zijn aan de verwachte frequenties. In formule (31) daarentegen wordt eerst een som gemaakt van de verschillen en deze som wordt gekwadrateerd. Stel dat er in een groep met twee scores, de ene score een verschil oplevert van -2 en de andere een verschil van +2. De som is 0 en de bijdrage aan X^2 voor deze groep is 0. Meer algemeen: positieve en negatieve verschillen tussen geobserveerde en verwachte frequenties kunnen elkaar compenseren voor ze gekwadrateerd worden. Theoretisch is het dus mogelijk dat een X^2 uitgerekend met formule (31) exact nul is, doch dat er binnen de groepen grote (elkaar compenserende) verschillen bestaan tussen geobserveerde en verwachte frequenties. De X^2 van (31) is (bij benadering) chi-kwadraat verdeeld met r-1 vrijheidsgraden. Maar door de compenserende werking van formule (31) hebben we een toets gemaakt die ongevoelig is voor elkaar compenserende verschillen tussen geobserveerde en verwachte frequenties binnen de groepen. Een aardig theoretisch voordeel van deze aanpak is dat het niet uitmaakt hoe de groepen gedefinieerd worden. Toepassing van (31) levert steeds een grootheid op die chi-kwadraat

verdeeld is. We moeten er alleen rekening mee houden dat de som van de verwachte frequenties (zowel voor een juist als voor een fout antwoord) in de groepen niet te klein is. Verder zijn we vrij in de definitie van de groepen. (Ze mogen elkaar zelfs overlappen, doch dan is een veel ingewikkelder formule nodig dan (31) en die is niet in OPLM ingebouwd.) De groepsindeling gebeurt in OPLM automatisch met de volgende criteria:

a) Maak zoveel mogelijk groepen, maar niet meer dan 8.

b) Bij een volledig design zijn de scores die in een groep worden opgenomen opeenvolgend. (Het kan dus niet zo zijn dat score 11 en 13 tot één groep behoren en 12 en 14 tot een andere.)

c) In elke groep is de som der verwachte frequenties voor een juist antwoord minstens 5; idem voor een fout antwoord.

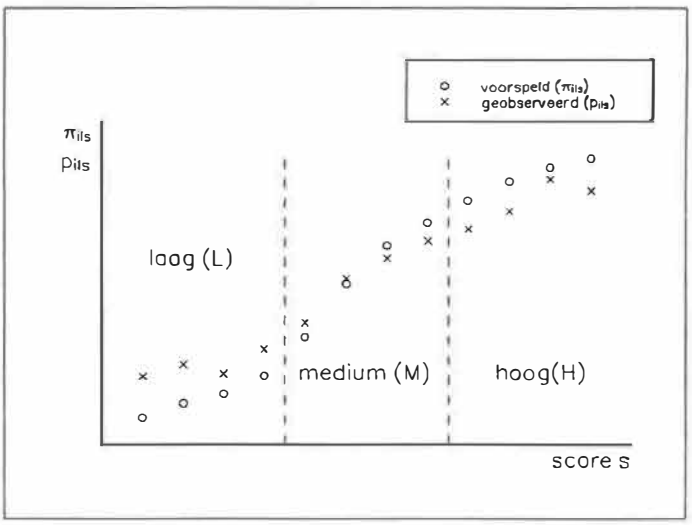
d) Elke groep is gebaseerd op minstens 30 observaties.

e) Er wordt naar gestreefd de groepen even groot te maken in termen van het aantal observaties. Strikte gelijkheid is gewoonlijk niet te bereiken, omdat alle personen met dezelfde score in dezelfde groep moeten worden opgenomen.

5.2.2 De M-toetsen

In de vorige subparagraaf hebben we gezien dat de X^2-toetsen ongevoelig zijn voor inbreuken tegen het model die elkaar binnen groepen van scores kunnen compenseren. In hoofdstuk 4 hebben we gezien dat in OPLM de discriminatie-indices per hypothese worden ingevoerd. Meestal weten we niet of de discriminatie-index die met een bepaald item geassocieerd is, hoog moet zijn of laag. Als we zo maar wat invullen, is er veel kans dat we een verkeerde waarde kiezen voor a_i. Veronderstel dat we die waarde te hoog hebben gekozen. In zo'n geval zullen de theoretische probabiliteiten een steiler verloop vertonen dan de geobserveerde proporties (die natuurlijk onafhankelijk zijn van onze hypothese). Een (beetje geflatteerd) voorbeeld is weergegeven in Figuur 19. Stel dat we de scores verdelen over drie groepen zoals weergegeven in de figuur. In de L-groep is het verschil $p_{i|s} - \pi_{i|s} > 0$, zodat er geen compensatie optreedt door het groeperen. In de H-groep is het verschil systematisch kleiner dan 0, zodat ook hier geen compensatie optreedt. In de M-groep treedt er wel compensatie op. Dat wil

zeggen dat door het toepassen van formule (31) de afwijkingen tussen verwachte en geobserveerde frequenties goeddeels zullen worden ontdekt. De X^2-toets is dus gevoelig voor het overschatten van de discriminatie-index. Indien de discriminatie-index onderschat wordt, krijgen we het omgekeerde beeld: de geobserveerde proporties zullen een steiler verloop vertonen dan de theoretische, maar de X^2-toets is er gevoelig voor. De X^2-waarde zal dus groot zijn bij onderschatting van a_1, bij overschatting, maar ook bij afwijkingen die geen mooi systematisch patroon vertonen, zoals in Figuur 18 (als de afwijkingen groot genoeg zijn natuurlijk om door de toets te worden ontdekt.) Het vinden van een grote (en significante) X^2-waarde geeft dus aan dat er iets met het model niet in orde is, doch uit de getalswaarde van X^2 kunnen we niet afleiden wat er aan de hand is. Een handig hulpmiddel is dan natuurlijk het tekenen van een figuur zoals Figuur 19. Dit is wat er gebeurt in het programma OPDRAW, met dien verstande dat niet alle scoregroepen afzonderlijk worden afgebeeld, doch alleen de gemiddelde proporties en probabiliteiten per groep van scores.



Figuur 19

Proporties en probabiliteiten wanneer a_1 overschat is

Omdat de specifieke oorzaak van de afwijking niet afgeleid kan worden uit de toetsingsgrootheid X^2, noemt men deze toetsen ook wel ongericht. Er

bestaat echter ook een mogelijkheid om gerichte toetsen te maken, dat wil zeggen toetsen die een andere uitkomst geven bij overschatting van de discriminatie-index dan wel bij onderschatting. De toetsingsgrootheid M wordt gedefinieerd als

$$M = \frac{\sum_{s \in L} n_s (p_{i|s} - \pi_{i|s}) - \sum_{s \in H} n_s (p_{i|s} - \pi_{i|s})}{N}. \qquad (32)$$

Uit wat hierboven gezegd is, zal het duidelijk zijn dat, voor het voorbeeld in figuur 19, de eerste som in de teller van (32) positief zal zijn en de tweede negatief. Het verschil zal dus positief zijn. Indien de discriminatie-index onderschat is zal de teller van (32) negatief zijn. De noemer N specificeren we hier verder niet. De functie van de noemer is er weer voor om te zorgen dat M een grootheid is die uit een bekende verdeling komt. Indien het model waar is, komt M uit een standaardnormale verdeling. Waarden van M die in absolute waarde groter zijn dan 1.96 (5%) of 2.58 (1%), zijn dus 'verdacht' groot. Grote positieve waarden wijzen op een overschatting van a_i, negatieve op een onderschatting. Op die manier weten we niet alleen dat er iets aan de hand is met het model, doch we krijgen ook een hint voor verandering: we kunnen een nieuwe analyse proberen met een a_i die aangepast is in de richting die door de M-toets is aangegeven. We hebben echter nog geen precieze definitie gegeven van wat een lage, een medium en een hoge score is. In het programma worden drie verschillende definities gehanteerd en voor elk item krijgen we dus drie M-toetsen, die respectievelijk de naam M, M2 en M3 dragen.

1 M-toets: $s \in L$ indien $\pi_{i|s} \leq 0.4$; $s \in H$, indien $\pi_{i|s} \geq 0.6$.
In woorden: scores waarvoor de kans op een goed antwoord op item i kleiner of gelijk is aan 40% behoren tot de laaggroep, en voor de hooggroep is de kans groter of gelijk aan 60%. De andere scores behoren tot de middengroep. Nu kan een item zo gemakkelijk zijn dat alle scores tot de H-groep behoren. In zo'n geval is M steeds exact gelijk aan nul, doch helemaal niet informatief. Op de uitvoer van OPLM verschijnt dan niet 0.0 als waarde maar 99.999. Let wel: voor

de M-toets is de verdeling in drie groepen afhankelijk van het item. Dit geldt ook voor de X^2-toetsen, doch niet voor de M2 en M3 toetsen.

2 M2-toets: de totale steekproef wordt in twee (ongeveer) gelijke helften verdeeld: de eerste helft bevat de personen met de laagste scores, de tweede helft de personen met de hoogste scores. De scores die voorkomen bij de eerste helft vormen de L-groep; de andere vormen de H-groep. De midden-groep is leeg.

3 M3-toets: analoog met M2, doch nu heeft er een driedeling plaats van de steekproef.

In Tabel 6 staan de resultaten van de statistische toetsen uit het voorbeeld dat in hoofdstuk 4 werd gebruikt (zie Tabel 1).

Tabel 6

Statistische toetsen horend bij de CML-analyse uit Tabel 1

X2*	DF	P	M*	M2*	M3*
5.056	4	.282	1.066	.893	-.008
2.510	4	.643	-.740	-.890	.108
1.329	4	.856	.781	.373	.951
2.506	4	.644	.167	.294	-.585
4.013	4	.404	.664	-.110	.383
1.744	4	.783	-.662	-.557	-.806

Geen enkele van de M, M2 of M3-waarden is in absolute waarde groter dan 1.96, en geen enkele van de X^2-waarden heeft een overschrijdingskans (P) kleiner dan 0.05. Er is dus geen enkele aanwijzing dat het model niet past. De kolom DF geeft per X^2-toets het aantal vrijheidsgraden aan. Uit de tabel kunnen we dus afleiden dat het programma voor elk item de scores heeft ingedeeld in DF+1 = r = 5 groepen. Het '*' in de kop van de tabel geeft aan dat de correctie voor het schatten van de parameters niet is toegepast.

Tenslotte nog enkele opmerkingen voor het verstandige gebruik van de statistische toetsen:

1 De toetsen voor de verschillende items zijn niet onafhankelijk van elkaar. Een M-toets kan significant zijn, omdat bijvoorbeeld de a-waarde van het item veel te hoog is. De M-toets voor dat item is daar speciaal gevoelig voor. Doch significantie kan ook optreden, omdat er iets niet in orde is met een ander item. Daarom moet men voorzichtig zijn met het aanpassen van de a-waarden. Het verdient aanbeveling slechts een klein aantal veranderingen tegelijk aan te brengen, bij voorkeur bij de items met de (absoluut) grootste M, M2 of M3 waarden en/of met de kleinste overschrijdingskans, en dan de analyse opnieuw uit te voeren.

2 De drie M-toetsen zijn bedoeld om gevoelig te zijn voor dezelfde fout in het model. Ze hebben natuurlijk niet alle drie precies hetzelfde onderscheidend vermogen, zodat het kan voorkomen dat bijvoorbeeld M en M2 significant zijn, doch M3 niet. Als M en M2 dan hetzelfde algebraïsch teken hebben, wijst dit op een misspecificatie van de a-waarde, zeker als ook de X^2-toets significant is. Het kan echter gebeuren dat twee of drie M-toetsen significant zijn, doch niet met hetzelfde algebraïsch teken. In zo'n geval kan het model meestal niet aangepast worden door een andere a te kiezen, maar is er (waarschijnlijk) iets anders aan de hand. Het verdient altijd aanbeveling de tabel met statistische toetsen te bestuderen in samenhang met de grafische uitvoer die OPDRAW aanmaakt.

3 Een significante toets betekent niet 'het bewijs' dat het model niet goed is. Indien het model goed is en de toetsen onafhankelijk zouden zijn, dan mag men verwachten dat 5% van de toetsen significantie op het 5%-niveau zullen opleveren, of algemener: de overschrijdingskansen, P-waarden, moeten een uniforme verdeling hebben. Hoewel de toetsen niet strikt onafhankelijk zijn van elkaar, is de afhankelijkheid niet erg groot en een min of meer uniforme verdeling van de P-waarden kan geredelijk als een acceptabele uitkomst worden beschouwd. Gaat men echter koppig verder met aanpassingen van het model tot er geen enkele significantie meer optreedt, dan 'kapitaliseert men op toeval'. Dit wil zeggen dat men het model tot in de details gaat aanpassen aan de eigenaardigheden van de steekproef die men voorhanden heeft. (Dit

resulteert meestal in een model waar men grote a-waarden moet kiezen om ook kleine nuances in discriminatie 'te vangen'.) Gaat men dan echter deze a-waarden gebruiken om gegevens van een andere steekproef (met andere eigenaardigheden) te analyseren, dan krijgt men vaak tegenvallende resultaten.

4 Denk bij de analyse ook aan fouten van de tweede soort: geen significantie betekent niet automatisch dat het model goed is. Er kunnen defecten in het model zitten die door de toets niet ontdekt worden, bijvoorbeeld omdat de steekproef te klein is. Gaat men dan later hetzelfde model toepassen op een grotere steekproef, dan kan blijken dat er plots heel veel toetsen significant worden. Het onderscheidend vermogen van de toetsen is niet alleen afhankelijk van de steekproefgrootte, maar ook van de p-waarde van het item. Bij items met een extreme p-waarde is het onderscheidend vermogen van de toetsen niet groot: verkeerde a-waarden worden niet gemakkelijk ontdekt. Geeft men later dezelfde items aan een steekproef uit een populatie waar de p-waarde minder extreem is, dan kunnen die fouten ineens wel tevoorschijn komen.

5 De hierboven besproken toetsen kunnen alleen uitgevoerd worden, als de analyse gebeurt met de CML-schattingsmethode (OPCML).

5.3 De R_1-toets

Het kan natuurlijk voorkomen dat geen enkele van de itemgerichte X^2-toetsen significant is, maar dat het resultaat ook niet erg bevredigend is. Bijvoorbeeld indien bijna geen enkele P-waarde groter is dan 0.5. De R_1-toets is een zogenaamde 'overall' toets die de afzonderlijke itemgerichte toetsen combineert. Indien de X^2-toetsingsgrootheden onafhankelijk van elkaar zouden zijn, dan zou hun som ook chi-kwadraat verdeeld zijn. Omdat ze niet onafhankelijk zijn, moeten er speciale voorzorgsmaatregelen getroffen worden. Formule (33) hieronder geeft weer hoe de toetsingsgrootheid R_i berekend wordt. (Het betreft weer een benaderende

formule, die geen rekening houdt met het feit dat de itemparameters uit de data geschat zijn).

$$R_1 = \sum_{i=1}^{k} \sum_{q=1}^{r} \frac{\left[\sum_{s \in G_q} n_s (p_{i|s} - \pi_{i|s})\right]^2}{\sum_{s \in G_q} n_s \pi_{i|s} (1 - \pi_{i|s})} \qquad (33)$$

Vergelijking van (33) met (32) suggereert dat (33) niets anders is dan de som van de X^2-grootheden per toets. Er is echter een belangrijke restrictie: bij de X^2-toetsen zagen we dat de groepering van scores in groepen bij ieder item opnieuw gebeurt om ervoor te zorgen dat er geen kleine verwachte frequenties ontstaan. Bij formule (33) kan dit niet: dezelfde groepering moet gelden voor alle items. (Indien ongelijke groeperingen gebruikt worden, ontstaat er een formule die veel ingewikkelder is en die bijna niet meer uitgerekend kan worden.) In het programma worden de X^2-grootheden niet zomaar bij elkaar opgeteld. Er wordt een nieuwe gemeenschappelijke groepering gemaakt (in ten hoogste 4 groepen) die gebruikt wordt om (33) uit te rekenen. Daarbij is het in vele gevallen onvermijdelijk dat er kleine verwachte frequenties ontstaan. Deze kleine verwachte frequenties worden geteld en indien hun aantal te groot is, is een beroep op de theoretische chi-kwadraatverdeling niet meer gerechtvaardigd. Het is echter niet zonder meer duidelijk wat verstaan moet worden onder 'te groot'. Bij de gewone analyse van contingentietabellen wordt de regel aangehouden dat het aantal cellen met een verwachte frequentie kleiner dan 5 niet groter mag zijn dan 20% van het aantal cellen. Dit is echter een zeer ruwe vuistregel, die niet blind moet worden toegepast. Het gebruik van de R_1-toets is dus een beetje problematisch. We komen hierop terug bij de bespreking van statistische toetsen bij onvolledige designs. In Tabel 7 staat de uitvoer die OPCML geeft met betrekking tot de R_1-toets. Het betreft het voorbeeld dat in Tabel 1 en Tabel 6 reeds is besproken. De uitkomst van formule (33) op de gegevens staat vermeld onder het kopje 'sum sq' (sum of squares); past men de correctie toe voor het schatten van de parameters, dan krijgt men twee uitkomsten: 'sum sq' en de gecorrigeerde (en te preferen) uitkomst onder het kopje 'R_{1c}'. (De c is bijgevoegd omdat het gaat over de R_1-toets met CML-schatters. Er is ook een analoge toets, indien men werkt met MML (R_{1m}),

die we hier niet bespreken.) De grootheid '#deviates' geeft het aantal vrije cellen aan; het aantal vrijheidsgraden is het aantal vrije cellen minus het aantal geschatte itemparameters. Bij ingewikkelde designs is het bereken van '#deviates' soms nogal ingewikkeld. Het programma draagt zorg voor de berekening. Onder de tabel wordt een overzicht gegeven van het aantal cellen dat een 'kleine' verwachte frequentie heeft. Bij dichotome items betekent 'nonzero category' niets anders dan 'juist antwoord'. Bij polytome items zijn het de categorieën die een niet-nul score hebben opgeleverd. Omdat er in dit voorbeeld slechts 6 items werden gebruikt en 1000 observaties, zijn de verwachte frequenties groot genoeg om vertrouwen te kunnen hebben in de R_1-toets.

Tabel 7

Uitvoer van OPCML met betrekking tot de R_1-toets

SUMMARY OF R1c-STATISTICS

booklet	#items	#groups	#deviates	sum sq.	R1c
1	6	4	20	13.46	11.53

R1c = 11.534; df = 15; p = .7139

Number of deviates for nonzero categories = 24
Number of cells with expected frequency < 5 = 0 (.0%)
Number of cells with expected frequency < 1 = 0 (.0%)

Number of deviates for zero categories = 24
Number of cells with expected frequency < 5 = 0 (.0%)
Number of cells with expected frequency < 1 = 0 (.0%)

Number of deviates for all categories = 48
Number of cells with expected frequency < 5 = 0 (.0%)
Number of cells with expected frequency < 1 = 0 (.0%)

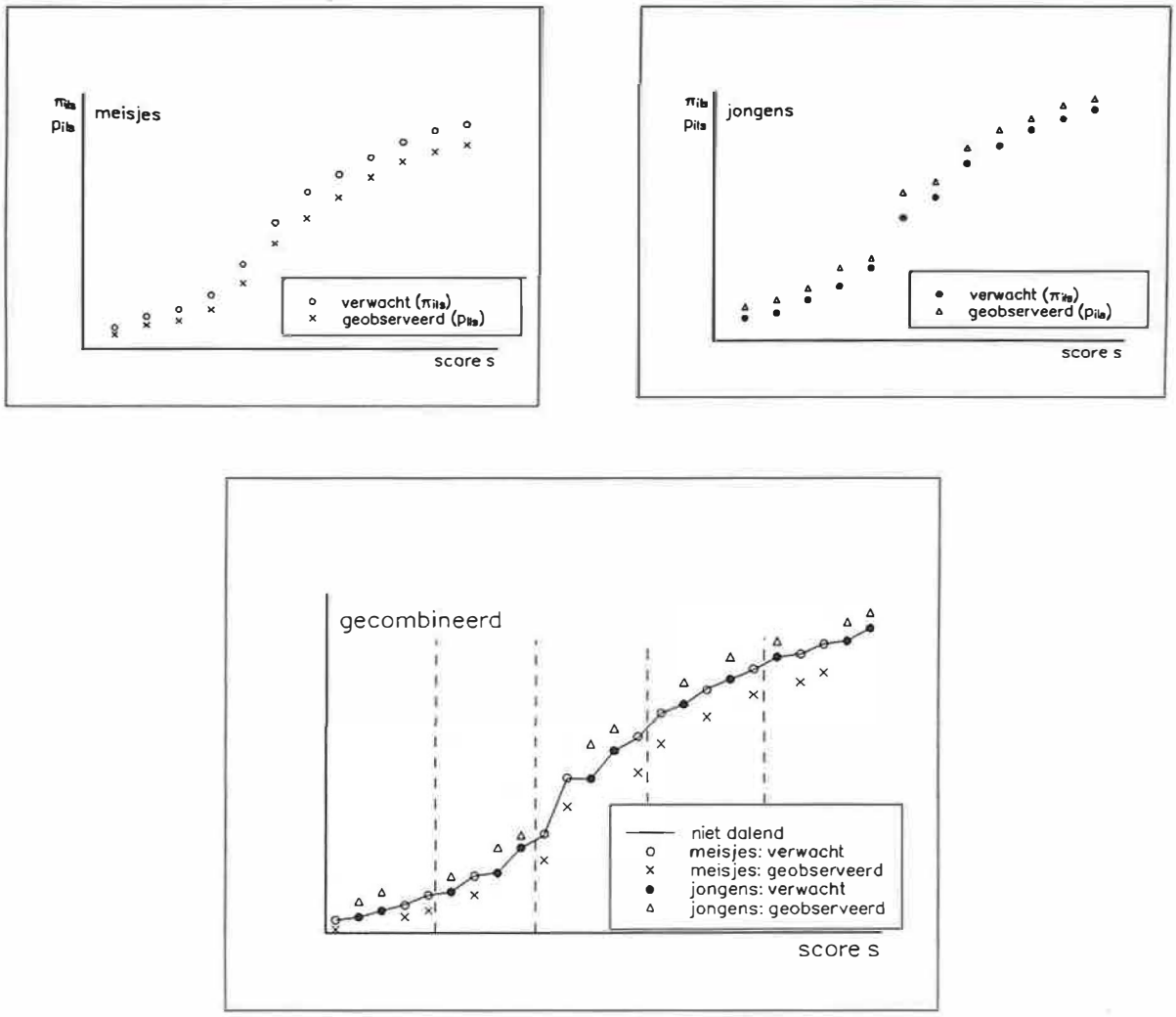
Indien er gewerkt wordt met een onvolledig design, is de veralgemening voor de R_1-toets erg eenvoudig: formule (33) wordt uitgerekend voor elk boekje afzonderlijk en de R_1-toetsingsgrootheid is gewoon de som van R_1-grootheden per boekje. Deze grootheden, die wel gerapporteerd worden in de uitvoer, kunnen echter niet als chi-kwadraat verdeelde variabelen geïnterpreteerd worden, omdat de parameters geschat zijn uit alle boekjes samen. Deze

waarden kunnen wel gebruikt worden voor een benaderende interpretatie: bij een acceptabel model moet voor elk boekje gelden dat de R_{1c}-bijdrage (of de 'sum sq') van dezelfde grootorde is als '#deviates'. Zit er nu een boekje bij waar de R_{1c}-bijdrage veel groter is dan '#deviates', terwijl dit niet zo is bij de andere boekjes, dan is het zinvol de modelafwijking te zoeken in eigenaardigheden van dit boekje: één of meerdere items die alleen in dat boekje voorkomen, of de steekproef waarbij dat boekje is afgenomen, als bijvoorbeeld één of twee scholen om wat voor reden dan ook de items niet 'serieus' hebben beantwoord. Het is dan ook ten zeerste aan te raden met eenvoudige analysetechnieken (TIA bijvoorbeeld) de gegevens te screenen, vooraleer met OPLM te gaan werken. Het uitzoeken van zulke methodes en het correct interpreteren vergt echter wel enige methodologische ervaring.
De itemgerichte X^2-toetsen bij onvolledige designs worden in de volgende paragraaf behandeld.

5.4 Toetsen op item-bias

Om de toetsen voor item-bias goed te begrijpen is het belangrijk in te zien hoe de itemgerichte X^2-toetsen geconstrueerd worden bij onvolledige designs. Veronderstel dat een bepaalde toets met 13 items, aan een steekproef van jongens gegeven wordt, en een andere toets, eveneens met 13 items aan een steekproef meisjes. De toetsen hoeven niet precies dezelfde items te bevatten, doch we veronderstellen dat enkele items gemeenschappelijk zijn aan beide toetsen. Item i is zo'n gemeenschappelijk item. Veronderstel nu dat item i gebiased is ten voordele van de jongens. Bij de schatting worden de gegevens van jongens en meisjes samengevoegd en het programma berekent de moeilijkheidsgraad van item i als een soort compromis tussen de moeilijkheidsgraad in de jongenspopulatie en de meisjespopulatie. Als we echter plaatjes maken met de conditionele probabiliteiten en proporties, dan krijgen we een beeld zoals in de bovenste twee plaatjes van Figuur 20: de jongens doen het systematisch beter dan verwacht en de meisjes slechter.
De standaardmanier waarop de X^2-toets uitgerekend wordt in OPCML gaat als volgt. Omdat boekjes een verschillende samenstelling kunnen hebben, zijn de

scores over de boekjes heen niet rechtstreeks vergelijkbaar met elkaar: een lage score in een moeilijk boekje kan duiden op een grotere vaardigheid dan een hoge score in een gemakkelijk boekje. De manier waarop binnen OPLM twee scores uit verschillende boekjes met elkaar vergeleken worden gaat via de conditionele probabiliteit $\pi_{i|s}$.



Figuur 20

De constructie van de itemgerichte X^2-toets in onvolledige designs

In het onderste gedeelte van Figuur 20 zijn de bovenste twee plaatjes zodanig in elkaar geschoven dat de theoretische probabiliteiten monotoon

stijgen van links naar rechts. (De lijn die de cirkeltjes verbindt daalt nergens.) Deze ordening is uniek, uitgezonderd in het geval dat twee of meer probabiliteiten exact aan elkaar gelijk zijn.
In dat geval beslist het toeval over de ordening. Op basis van deze gecombineerde curve gebeurt de groepering van scores (aangegeven door de verticale stippellijnen). Uit Figuur 20 blijkt duidelijk dat er binnen de groepen, waarin nu zowel meisjes als jongens zijn vertegenwoordigd, compensatie zal optreden tussen positieve en negatieve afwijkingen, met als gevolg dat de X^2-toets niet erg gevoelig zal worden voor de itembias. De remedie daartegen is dan ook voor de hand liggend: als we het programma verhinderen om de twee curves in elkaar te schuiven en de groepering maken op de gegevens van jongens en meisjes afzonderlijk (dus in de bovenste twee plaatjes van Figuur 20), dan zal die compensatie niet optreden: bij de jongens zal in elke groep de gemiddelde geobserveerde frequentie groter zijn dan de verwachte, en bij de meisjes kleiner. Het programma berekent dus formule (31) voor de jongens en de meisjes afzonderlijk. De som geeft de X^2-toetsingsgrootheid en het aantal vrijheidsgraden is gelijk aan het aantal groepen jongens plus het aantal groepen meisjes minus 2.
Het programma OPCML biedt nog wat meer mogelijkheden voor onderzoek naar itembias. Stel dat er zes boekjes zijn, drie die aan jongens gegeven zijn en drie aan meisjes. Om bias met betrekking tot het geslacht te onderzoeken, laten we toe dat het programma de gegevens van de jongens afzonderlijk mag combineren, en die van de meisjes ook, doch de gegevens van jongens en meisjes worden niet gecombineerd. Het aansturen van het programma gaat als volgt. In scherm 2 (Design; zie Figuur 12) geeft men in het veld '#Stats' (= aantal statistische groepen) 2 op, omdat de variabele geslacht maar twee waarden heeft. In het kadertje dat dan verschijnt kan men de labels voor deze waarden opgeven (bijvoorbeeld: 1 = jongens; 2 = meisjes). In het subscherm waarin de boekjesspecificatie wordt opgegeven, geeft men in het veld 'staT' op, of het boekje aan jongens (1) of aan meisjes (2) is gegeven. Het biasonderzoek kan dus alleen dan uitgevoerd worden, indien de variabele waarvoor men bias onderzoek doet constant is voor alle personen die hetzelfde boekje hebben gekregen. Indien dit niet zo is, dient men kunstmatig nieuwe boekjes aan te maken.
Voorbeeld: Stel dat de data verzameld zijn in een volledig design. In kolom 1 van het gegevensbestand staat een '1' (boekjesnummer), en in kolom 2

staat het geslacht, gecodeerd als respectievelijk '1' en '2'. De items staan in de kolommen 3 t/m 20. Het format om de data te lezen zal dus zijn: '(I1,1X,18I1)'. Wil men biasonderzoek doen met betrekking tot de variabele geslacht, dan deelt men OPIN mee dat er twee boekjes zijn en dat het boekjesnummer in kolom 2 staat (Format: '(1X,19I1)'). Voor de schattingen maakt dit geen enkel verschil, doch de toetsen worden gevoelig voor eventuele bias. (Het verdient wel aanbeveling in zo'n geval het databestand te sorteren op geslacht, doch noodzakelijk is dit niet.)
Hoewel geslacht een beetje een schoolvoorbeeld is van een variabele waarop men biasonderzoek doet, kan natuurlijk elke andere variabele gebruikt worden om biasonderzoek te doen. Men zou bijvoorbeeld leeftijd kunnen nemen. Hoewel leeftijd een continue variabele is, dient men de respondenten te groeperen in homogene groepen naar leeftijd. Het maximale aantal waarden dat de biasvariabele mag aannemen is 4.
OPDRAW levert dan plaatjes op, die vergelijkbaar zijn met de bovenste twee plaatjes van Figuur 20 (na groepering).

6 Het programma-pakket OPLM (II)

Dit hoofdstuk is het vervolg van hoofdstuk 3. In paragraaf 1 komen de opties van OPIN aan de orde die in hoofdstuk 3 niet werden besproken. Een goed begrip van deze opties vereist dat kennis is genomen van de hoofdstukken 4 en 5. Één optie ('maXs') wordt ook besproken in hoofdstuk 7 waarin het model voor polytome items wordt behandeld.
In paragraaf 2 wordt de uitvoer van OPCML behandeld aan de hand van een voorbeeld.
Paragraaf 3 bevat een bespreking van het programma OPSUG waarmee redelijke schattingen kunnen worden gemaakt van de discriminatie-indices.
In paragraaf 4 wordt het programma OPREAD behandeld waarmee parameterschattingen uit het PAR-bestand zonder nauwkeurigheidsvelies kunnen worden ingelezen als 'fixed' woorden in het SLR-bestand.
In paragraaf 5 wordt het programma OPSIM besproken. Met dit programma kunnen artificiële data worden aangemaakt volgens het algemene OPLM-model.

6.1 Het programma OPIN (vervolg)

6.1.1 Opties via OPIN

Het programmapakket OPLM is geschikt voor zowel dichotome als polytome items. Een dichotoom item is een item waarmee ofwel één ofwel nul punten kunnen worden verdiend. 'Nul' en 'één' worden de categorieën van het item genoemd. Een meerkeuzevraag met vier alternatieven die dichotoom gescoord wordt (juist of onjuist), is dus een dichotoom item. Hoewel natuurlijk interessante onderzoekingen zijn te doen naar het keuzegedrag van de respondenten met betrekking tot de vier alternatieven, dient benadrukt te worden dat OPLM hiervoor niet geschikt is. Als we binnen het OPLM-model spreken over polytome items, dan wordt daarmee bedoeld items waarop in meer dan twee geordende categorieën kan worden gescoord. Deze categorieën kan men het beste interpreteren als 'het aantal punten' (de itemscore) dat met een antwoord is behaald. Essentieel is de interpretatie dat een hogere itemscore een aanduiding is van een grotere vaardigheid dan een lagere score. (Dit is de betekenis van de eis dat de categorieën geordend zijn.)

De scores die men kan behalen op een item zijn 0, 1, 2, ..., dus alleen opeenvolgende gehele aantallen; het minimum is nul.

maXs. (1-9) Het maximum aantal punten dat op een item behaald kan worden. Voor dichotome items is dit gelijk aan 1. Dit maximum hoeft niet voor ieder item hetzelfde te zijn. Men dient in toepassingen dit maximum niet lichtvaardig op te schroeven: het aantal parameters dat per item geschat wordt, is gelijk aan maXs. B vendien moet elke categorie in de data minstens één keer voorkomen, anders is het model niet te schatten.

paRs. (Free/fiXed) Free betekent dat de parameters die met het item geassocieerd zijn uit de data geschat dienen te worden. fiXed betekent dat die parameters bekend zijn en door het programma als bekende constanten behandeld dienen te worden. Kiest men deze optie, dan verschijnt een klein subschermpje waarin de waarden van die parameters moeten worden ingevuld. Commentaar op deze optie wordt gegeven bij de bespreking van het programma OPREAD.

> Eén toepassing kan erg handig zijn. Het programma OPCML laat het nulpunt van de schaal samenvallen met het middelpunt van de itemparameters. Men kan een ander nulpunt kiezen door een enkele itemparameter te fixeren. Vult men '0' in voor de waarde van de parameter van een dichotoom item, dan valt het nulpunt van de schaal samen met die parameter van dat item. Gefixeerde parameters zijn constanten, en hebben dus een standaardfout van nul. In de uitvoer wordt dit aangegeven met '---'. De standaardfouten van de andere parameters zullen in de regel groter zijn dan bij de normering die OPCML en OPMML gebruiken.

Als bij een polytoom item parameters gefixeerd worden, moet een waarde opgegeven worden voor alle parameters van het item.

Item tSts. (Complete/Diagonal) De optie Complete geeft aan dat voor de item-gerichte X^2-toetsen, de M-toetsen en de R_1-toets, de correctie voor het schatten van de parameters wordt uitgerekend. (Technisch: er wordt gebruik gemaakt van de gehele informatiematrix.) De optie Diagonal geeft aan dat van die correctie geen gebruik wordt gemaakt. (Technisch: de optie D gebruikt alleen de hoofddiagonaal van de informatiematrix.) (De formules (31), (32) en (33) worden gebruikt.) De optie D is veel sneller dan de

optie C, doch geeft slechts benaderende resultaten. Hoe goed die benadering is in het algemeen, is niet bekend. Ervaring met OPLM heeft geleerd dat de te gebruiken benaderingen in veel gevallen goed zijn.

De optie die hier aangegeven wordt, is niet onafhankelijk van de optie in het veld Est proc. (schattingsprocedure). Wil men bij het toetsen de complete methode gebruiken, dan moeten de noodzakelijke tabellen (o.a. de informatiematrix) aangemaakt worden tijdens de laatste fase van de schattingsprocedure. Dit kan alleen indien men voor de schattingsprocedure eveneens de optie Complete heeft opgegeven. De combinatie 'Est proc.: D/item tSts: C' wordt dan ook door OPIN geblokkeerd.

Ervaring heeft uitgewezen dat de berekeningen in het algemeen het snelst verlopen met de combinatie 'Est proc.: C/item tSts: D'. Indien verschillende analyses gedaan moeten worden op dezelfde data, is deze combinatie dan ook aan te raden. Meestal kan volstaan worden met een enkele nauwkeurige berekening (combinatie C/C) in de eindfase van de analyse.

Indien het aantal te schatten parameters groter is dan 181, wordt automatisch de combinatie D/D uitgevoerd.

siZe. (Standard/Extended/Huge). Hoeveelheid uitvoer. De opties zijn genest veld: Extended bevat (en wordt voorafgegaan door) Standard, Huge bevat (en wordt voorafgegaan door) Extended (alleen voor OPCML).

Standard bestaat uit de volgende delen:
- Echo van de inputopties;
- Resultaten van de klassieke toets- en itemanalyse, indien gevraagd;
- Centrale tabel met parameterschattingen, standaardfouten en statistische toetsen;
- Algemene statistische informatie, onder meer een summiere tabel van de R_1-toets (zie Tabel 7).

Extended bevat daarenboven
- Tabellen met geobserveerde en verwachte frequenties voor de itemgerichte X^2-toetsen en M-toetsen;
- Tabellen met gestandaardiseerde afwijkingen voor de R_1-toets en opsplitsingen van de kwadratensom en bijdragen aan de R_1-toets voor de verschillende boekjes, items en scoregroepen.

Huge bevat daarenboven tabellen met geobserveerde en verwachte frequenties voor de R_1-toets.

#Stats (1-4) Het aantal statistische groepen voor de berekening van de itemgerichte X^2-toetsen (alleen OPCML). Elk boekje behoort tot één statistische groep. Boekjes die tot dezelfde statistische groep behoren worden 'gemengd' (zie hoofdstuk 5) vooraleer de groepsindeling wordt toegepast. De verschillende statistische groepen worden nooit gemengd, zodat er geen compensatie van positieve en negatieve afwijkingen kan optreden door groepering. Deze optie is geschikt om onderzoek naar itembias te doen. (Zie ook het veld 'staT'.)

Voorbeeld: Stel dat er zes boekjes zijn, met boekjes 1, 2 en 6 aangeboden aan jongens en boekjes 3, 4 en 6 aangeboden aan meisjes. Wil men de gegevens van jongens en meisjes niet 'mengen' voor de berekening van de itemgerichte X^2-toetsen, dan geeft men hier '2' op. Is het gekozen aantal groter dan 1, dan verschijnt er een subschermpje waar men namen voor de verschillende statistische groepen kan opgeven. (Vervolg voorbeeld bij het veld 'staT'.)

staT. Voor elk boekje afzonderlijk moet worden opgegeven tot welke statistische groep dit boekje behoort. Het opgegeven getal mag niet groter zijn dan bij '#Stats' is opgegeven. (Zie aldaar.)

Voorbeeld (vervolg): Veronderstel dat men 'jongens' heeft opgegeven voor de eerste groep en 'meisjes' voor de tweede groep. Bij de boekjes 1, 2 en 6 geeft men '1' op en bij de andere boekjes '2'. Controle op de juiste keuze is mogelijk met @V: de namen die geassocieerd is met de gemaakte keuze wordt zichtbaar.

6.1.2 Veranderen van opties met schakelaars

Een aantal opties die men in OPIN kenbaar heeft gemaakt, hebben hun effect bij uitvoering van het programma OPPRE (bijvoorbeeld toets- en itemanalyse), andere zijn alleen effectief bij aanroep van OPCML of OPMML. Voor deze laatste opties dient het programma OPPRE als doorgeefluik. Wil men in OPIN een optie veranderen die alleen voor OPCML of OPMML belangrijk is, dan dient toch weer OPPRE met de veranderde opties te worden

uitgevoerd. Om deze omweg te vermijden, kan men ook een aantal schakelaars meegeven bij aanroep van OPMML en OPCML. Deze schakelaars overschrijven de opties die in OPIN zijn opgegeven, doch ze veranderen de opties in het .SCR bestand niet; ze zijn alleen effectief indien ze bij de aanroep worden meegegeven. De volgende schakelaars zijn geïmplementeerd:

/PP{Y|N} : persoonsparameters (Yes/No)
/ES{C|D} : schattingsprocedure (Complete/Diagonal)
/TS{C|D} : toetsingsprocedure (Complete/Diagonal)
/LG{Y|N} : .LOG-bestand (Yes/No)
/PF{Y|N} : .PAR-bestand (Yes/No)
/CV{Y|N} : .COV-bestand (Yes/No)
/SZ{S|E|H} : omvang uitvoer (Standard/Extended/Huge)
/TA{1|2} : formaat tabellen (1 kolom/2 kolommen)
/R1{Y|N} : R1-toets (Yes/No); default = Yes (OPCML en OPMML)
/R0{Y|N} : R0-toets (Yes/No); default = Yes (alleen OPMML)

- De laatste twee schakelaars zijn niet te bedienen via OPIN. (De R0-toets toetst of de geobserveerde scorefrequenties goed overeenkomen met de verwachte scorefrequenties. Deze toets is in dit rapport niet besproken.)

- Een schakelaar begint met een slash ('/') of een min-teken '-'. Schakelaars moeten van elkaar en van andere argumenten in de commandolijn gescheiden worden door één of meer spaties. Hoofdletters en kleine letters mogen door elkaar worden gebruikt worden.

- De volgorde van de schakelaars in de commandolijn is willekeurig; ook de volgorde met betrekking tot JOBN is willekeurig. Het commando

  OPCML /SZH example -LGN [CR]

  Begin met de CML-analyse van het bestand EXAMPLE.PRE. Voor de uitvoer is de optie 'HUGE' geldig en er wordt geen .LOG-bestand aangemaakt.

6.2 Uitvoer van OPCML

6.2.1 Het .CML-bestand

Hierna volgt een voorbeeld van de uitvoer van OPCML. Het gaat om de analyse van de antwoorden van 1000 personen op zeven items. De data zijn aangemaakt met het programma OPSIM (zie paragraaf 6.5) volgens het Rasch-model en vervolgens met OPCML geanalyseerd. Hier en daar is wat commentaar aan de uitvoer toegevoegd. Dit commentaar is *cursief* afgedrukt.

```
Rasch analyse met zeven items  (Titel als ingevoerd in OPIN)
Name = VOORBLD                                    OPLM, Version 3.0


Comments.
---------

Eerste analyse (Commentaar ingevoerd in OPIN)

Control flow, options and defaults.
-----------------------------------
The data are read from DATA

With FORMAT
(8I1)


Number of (legal) records skipped    =  0 (alle boekjes stonden 'on')
Number of illegal records (skipped) =  0 (alle records hadden een
                            boekjesnummer dat in OPIN gespecificeerd was.)


OPPRE is called with argument C:\OPLM\CURSUS\VOORBLD.SCR


Logical variables: (T staat voor 'true', F voor 'false'.)
          (Deze logische variabelen komen overeen met keuzen uit OPIN,
          tenzij ze overschreven werden door schakelaars. De laatste
          logische variabele wordt door het programma uitgerekend.

          Classical item-analysis: T
          Subject-parameters: T
          Output covariance-matrix: F
          Make log-file: F
          Output parameter-file: T
```

Datafile in expanded form: F
Compute full information matrix: T ***(d.i. Est proc. = Complete)***
Tests use information matrix: T ***(d.i. item tSts = Complete)***
Each category is observed at least once: T
(elke categorie van elk item moet minstens eenmaal voorkomen, anders zijn de parameters niet te schatten. Bij dichotome items betekent dit dat elk item minstens eenmaal juist en eenmaal fout moet zijn beantwoord. Indien dit niet zo is, geeft het programma OPPRE op het scherm aan voor welke items dit niet het geval was.)

De volgende grootheden geven de stopcriteria aan. Deze stopcriteria zijn opgeslagen in het bestand OPCML.DEF. Dit is een ASCII-bestand dat voor de gebruiker toegankelijk is en waar de stopcriteria veranderd kunnen worden. Het bestand zelf bevat een uitvoerige uitleg. Ook voor OPMML kunnen de stopcriteria door de gebruiker worden veranderd, via het bestand OPMML.DEF.

	IEP and .NOT.INF	IEP and INF	N-R
Stop-criterion:	.5000E-06	.5000E-02	.1000E-07
Max #iterations:	25	20	10

NREFRESH = 5
(In de Newton-Raphson-procedure wordt de informatiematrix om de vijf iteraties opnieuw uitgerekend. Dit aantal is in te stellen via OPCML.DEF. De informatiematrix wordt steeds opnieuw uitgerekend in de laatste iteratie.)

KEEPPRE = 1
(Het .PRE-bestand wordt bewaard na uitvoering van OPCML. Indien 0, wordt het .PRE-bestand vernietigd na uitvoering van OPCML. Instelbaar via OPCML.DEF.)

Size of output = huge
Number of columns in tables = 2

Creation date .SCR-file : 5-22-1992 at 15:27
Creation date .PRE-file : 5-22-1992 at 15:28

De toets-en itemanalyse wordt uitgevoerd per boekje. Voor elk item dat in een boekje voorkomt, worden drie grootheden uitgerekend:

1 *De p-waarde. Bij polytome items is de p-waarde gedefinieerd als het totaal aantal behaalde punten op het item gedeeld door het maximaal te behalen aantal punten.*

2 *De item-test correlatie rit(u), waarbij de testscore gedefinieerd is als het aantal items juist. 'u' staat voor 'unweighted', de items worden niet gewogen.*

3 *De item-test correlatie rit(w), waarbij de testscore de gewogen testscore is: de som van de discriminatie-indices van de correct beantwoorde items (bij dichotome items), of algemener, de som van de itemscores vermenigvuldigd met de discriminatie-index. Indien alle items dezelfde discriminatie-index hebben, geldt dat rit(u)=rit(w).*

Item analysis for booklet 1

item	p-value	rit(u)	rit(w)	item	p-value	rit(u)	rit(w)
1	.791	.464	.464	4	.517	.581	.581
2	.706	.488	.488	5	.368	.509	.509
3	.606	.545	.545	6	.321	.507	.507
				7	.214	.487	.487

Number of observations = 1000

Results based on raw (unweighted) scores
Mean = 3.523
S.D. = 1.649
Alpha = .532

Results based on weighted scores
Mean = 3.523
S.D. = 1.649
Alpha = .532

Corr(u,w) = 1.000

Alpha is Cronbachs α-coëfficiënt, en geeft een ondergrens aan voor de betrouwbaarheid van de toetsscore. Bij dichotome items is Cronbachs α identiek aan de KR20-coëfficiënt.
Corr(u,w) is de correlatie tussen gewogen en ongewogen scores.

De volgende tabel geeft voor populatie 1, de p-waarde, het aantal observaties en het aantal keren dat elke categorie van elk item gekozen is. In dit voorbeeld lijkt dit een beetje triviaal. Bij onvolledige designs kan dit echter nuttig zijn. Stel dat boekjes 1 en 2 gegeven zijn aan een steekproef jongens (populatie 1) en boekjes 3 en 4 aan een steekproef

meisjes (populatie 2). Dan maakt het programma twee tabellen aan, zodat voor items die aan beide populaties zijn voorgelegd een zinvolle vergelijking van de p-waarden mogelijk wordt. Definitie van de populaties gebeurt via de velden '#Margs' en 'marG' van scherm 2 in OPIN. De titel van de tweede tabel zou er dan als volgt uitzien:

Distribution of responses across booklets for marg. population 2 (meisjes):

Booklets: 3 4

Distribution of responses across booklets for marginal population 1:

Booklets: 1

item	label	a	N	P	0	1
1	tv= -1.5	1	1000	.791	209	791
2	tv= -1.0	1	1000	.706	294	706
3	tv= -0.5	1	1000	.606	394	606
4	tv= 0.0	1	1000	.517	483	517
5	tv= 0.5	1	1000	.368	632	368
6	tv= 1.0	1	1000	.321	679	321
7	tv= 1.5	1	1000	.214	786	214

De volgende tabel is de centrale tabel van de uitvoer. Voor commentaar wordt verwezen naar hoofdstuk 4 en 5. Voor de itemnamen is de werkelijke waarde (true value) ingevuld waarmee de data gemaakt zijn. Hoewel de data volgens het model gemaakt zijn, zien we toch significante toetsresultaten voor item 4. We hebben hier dus duidelijk te maken met een fout van de eerste soort.

RESULTS CALIBRATION.

nr	label	A	B	SE(B)	X2	DF	P	M	M2	M3
1	tv= -1.5	1	-1.539	.078	2.972	3	.396	-.059	-.652	-.560
2	tv= -1.0	1	-1.020	.070	9.950	3	.019	.306	.110	1.799
3	tv= -0.5	1	-.498	.066	4.904	3	.179	.296	-.665	.193
4	tv= 0.0	1	-.068	.065	13.969	3	.003	-1.398	.097	-2.786
5	tv= 0.5	1	.659	.067	2.819	3	.420	.987	1.567	1.386
6	tv= 1.0	1	.908	.070	.231	3	.972	1.189	.454	.269
7	tv= 1.5	1	1.558	.079	2.984	3	.394	.462	-1.314	-.269

log-likelihood = -2237.76 (-.2237760661D+04)
-2*log-likelihood = 4475.52 (.4475521323D+04)

De 'log-likelihood' is de logaritme van de maximale waarde van de aannemelijkheidsfunctie. Deze waarde wordt gebruikt in een bepaalde statistische toets die wij niet besproken hebben.

Number of parameters estimated = 6
Geometric mean of discrimination indices = 1.000

Dit geometrische gemiddelde wordt besproken in paragraaf 6.3.

Distribution of p-values for X2-tests.

```
0.-----.1-----.2-----.3-----.4-----.5-----.6-----.7-----.8-----.9-----1.
    2      1      0      2      1      0      0      0      0      1
```

Indien alle itemgerichte X^2-toetsen onafhankelijk zouden zijn, en het model waar, moeten de P-waarden een rechthoekige verdeling vormen in het (0-1)-interval. Met zeven items is het echter zeer moeilijk te beoordelen of dit zo is.

Total: 7

Door de gehanteerde criteria voor het groeperen van scores is het soms niet mogelijk meer dan één groep te vormen. In zo'n geval is de X^2-waarde per definitie gelijk aan nul, maar niet informatief. Daarom wordt een P-waarde van 99.999 gerapporteerd. Deze wordt uiteraard niet in het bovenstaande 'balkje' opgenomen, en ook niet meegeteld in het totaal.

De volgende tabel is uitvoerig in hoofdstuk 5 besproken.

SUMMARY OF R1c-STATISTICS

booklet	#items	#groups	#deviates	sum sq.	R1c
1	7	4	24	37.54	32.99

R1c = 32.989; df = 18; p = .0167

Number of deviates for nonzero categories = 28
Number of cells with expected frequency < 5 = 0 (.0%)
Number of cells with expected frequency < 1 = 0 (.0%)

Number of deviates for zero categories = 28
Number of cells with expected frequency < 5 = 0 (.0%)
Number of cells with expected frequency < 1 = 0 (.0%)

Number of deviates for all categories = 56
Number of cells with expected frequency < 5 = 0 (.0%)
Number of cells with expected frequency < 1 = 0 (.0%)

De volgende tabel rapporteert de geschatte ϑ-waarde die met elke score is geassocieerd (per boekje). ϑ-waarden zijn over de boekjes heen met elkaar vergelijkbaar. De kolom 'freq' geeft de frequentie aan van de score in het boekje; de kolom SE(th) is de standaard-fout. De kolom 'bias' geeft de systematische onzuiverheid aan van de schattingen. De interpretatie is nogal ingewikkeld. We illustreren haar aan de hand van de uitkomsten voor score = 0. De schatting van ϑ is -3.225. (Het is een soort aangepaste ML-schatting; zie verder.) Dit betekent natuurlijk niet dat een persoon met score 0 precies die ϑ-waarde heeft. Het is ook niet de 'gemiddelde' of verwachte ϑ-waarde van alle personen die een score van 0 behaald hebben. (Om zo'n verwachte waarde uit te rekenen dienen bijkomende veronderstellingen gemaakt te worden. Bijvoorbeeld de veronderstelling van een normale verdeling van de ϑ-waarden zoals met in de MML-schattingsprocedure gebeurt.) Het enige zinvolle wat we hier kunnen doen is nagaan wat er 'gemiddeld' zal gebeuren met een persoon die een echte ϑ-waarde van -3.225 heeft. Als we de toets een zeer groot aantal keren aan zo'n persoon konden afnemen (onder exact dezelfde condities; de persoon leert niet!) dan zouden we natuurlijk niet iedere keer een score 0 observeren. Immers met elke ϑ-waarde kunnen alle scores voorkomen. We verkrijgen dus een distributie van scores. Voor elke score kunnen we ϑ schatten (de schattingen zijn precies de ϑ-waarden die in de tabel staan), zodat we ook de distributie van de schattingen kennen. Het gemiddelde van die schattingen is gelijk aan de echte ϑ-waarde + de bias = -3.225 + 0.522 = -2.703. Dit betekent dus dat we een persoon met een lage ϑ gemiddeld genomen gaan overschatten. Het betekent echter niet dat personen die een geschatte ϑ-waarde hebben van -3.225 gemiddeld genomen een echte ϑ-waarde hebben van -3.225 - 0.522 = -3.747. Het programma berekent de onzuiverheid voor enkele 'echte' ϑ-waarden. De waarden die gekozen zijn zijn precies de waarden die in de kolom 'theta' staan afgedrukt. De getallen in de kolom 'theta' hebben dus een dubbele functie: enerzijds zijn het de schattingen van ϑ die horen bij de score in de linkerkolom, anderzijds fungeren ze als de 'echte' ϑ's, waarvoor de onzuiverheid wordt gerapporteerd in de kolom 'bias'. De manier waarop de schatters bepaald worden (het zijn zogenaamde 'Warm'-schatters) zorgt ervoor dat de bias over het algemeen erg klein is. Alleen voor de extreme scores blijft hij aanzienlijk. (Technisch: de bias is gedefinieerd als
$E(\hat{\vartheta}|\vartheta) - \vartheta$.)

Estimates of latent abilities.

Distribution of scores and estimated latent abilities for booklet 1

score	freq.	theta	SE(th)	bias	score	freq.	theta	SE(th)	bias
0	24	-3.225	2.000	.522	4	216	.331	.852	-.001
1	95	-1.847	1.089	.061	5	155	1.022	.909	-.008
2	162	-1.022	.909	.008	6	95	1.846	1.089	-.061
3	219	-.330	.852	.001	7	34	3.224	2.000	-.522

Number of observations =	1000
Mean (theta's) =	.024
Variance (estimated theta's) =	1.629
Variance (true theta's) =	.594
Reliability (of est. theta's) =	.365
Correlation (est. th.,scores) =	.993

De bovenstaande waarden zijn alleen bruikbaar in de veronderstelling dat de steekproef van personen die het boekje heeft gekregen, een aselecte steekproef is uit de populatie waarin men geïnteresseerd is. Bovendien zijn de grootheden berekend in de veronderstelling dat de 'bias' overal nul is, of althans te verwaarlozen. De variantie van de 'ware theta's' is gedefinieerd als de variantie van de geobserveerde theta's minus de gemiddelde gekwadrateerde standaardfout. De betrouwbaarheid is gedefinieerd als de verhouding tussen 'ware' en 'geobserveerde' variantie. Simulatie-studies hebben uitgewezen dat zowel ware variantie als betrouwbaarheid meestal onderschat zijn. Deze uitkomsten moeten dus met de nodige voorzichtigheid gebruikt worden. Exacte theoretische resultaten zijn nog niet voorhanden. De laatste regel in bovenstaand tabelletje geeft de correlatie aan tussen geschatte θ-waarden en de scores op de toets.

EXTENDED OUTPUT

Tables for item-oriented X2-statistics

Number of observations (N), observed and expected number of correct responses (OBS(+) and EXP(+)) for general X2-tests.

Item 1. #contrasting groups = 1:

dichotomisation [0-0:1-1]: X2 = 2.972, DF = 3, P = .396

N:	257	219	216	250
OBS(+):	147	173	192	245
EXP(+):	146.2	176.9	193.6	240.3

Bovenstaande gegevens worden afgedrukt voor elk item. '#contrasting groups' betekent het aantal statistische groepen waar dit item aan is voorgelegd. Er geldt dus steeds: '#contrasting groups' ≤ '#Stats'. Zie hoofdstuk 7 voor de precieze betekenis van 'dichotomisation'. De tabel geeft aan dat de scores in vier groepen zijn ingedeeld. De eerste groep bevat 257 observaties, waarvan er 147 een correct antwoord waren. Het verwachte aantal correct is 146.2.

Tables for M-tests.

Number of observations (N), observed and expected number of correct responses (OBS(+) and EXP(+)).

Item 1.

dichotomisation [0-0:1-1]

	M		M2		M3	
Statistics:	-.059		.152		-.249	
	low	high	low	high	low	high
N:	0	847	476	466	257	250
OBS(+):	0	718	320	437	147	245
EXP(+):	.0	717.7	323.1	433.9	146.2	240.3

Net als bij de X^2-toetsen worden ook hier per item de frequenties aangegeven waarop de M-toetsen berekend worden. Voor de M-toets is het niet te vermijden dat er soms in de hoog-groep of in de laag-groep zeer kleine frequenties verschijnen, als gevolg van de definitie van de L- en H-groepen. Bij de M2-toets waar de middengroep per definitie leeg is, zien we dat niet alle observaties zijn opgenomen: 257 + 685 = 942 maar er waren 1000 observaties. De 58 ontbrekende observaties betreffen een nul-score of een perfecte score. Die worden niet in de M-toetsen betrokken. Bij de M3-toets verwachten we in de L-groep en de H-groep 942/3 = 314 observaties in plaats van de behoorlijk veel kleinere 257 resp. 250. Dit komt doordat het aantal verschillende scores klein is, en personen met dezelfde score in dezelfde groep opgenomen dienen te worden. Bij toetsen met meer items wordt de proportie van 1/3 meestal beter benaderd.

De precieze groepsindeling wordt niet gegeven, omdat dit bij onvolledige designs met veel boekjes tot zeer uitgebreide en onoverzichtelijke uitvoer leidt. Bij de volgende tabel gebeurt dit wel omdat daar slechts één groepsindeling gehanteerd wordt.

Tables for R1c-statistic
==========

Booklet 1
==========

groups:	1	2	3	4	
range:	1 - 2	3 - 3	4 - 4	5 - 6	
#obs.:	257	219	216	250	
item cat					
1 1	.11	-.67	-.36	1.54	2.96
2 1	.51	-.49	1.69	-2.56	9.91
3 1	1.04	-1.54	-.23	1.17	4.88
4 1	-2.44	2.12	-1.21	1.39	13.85
5 1	1.16	.67	-.53	-.81	2.73
6 1	.10	.38	-.13	-.23	.23
7 1	-.93	-.74	1.24	-.13	2.98
sum sq.:	9.53	8.71	6.34	12.96	37.54
R1c:	8.33	6.94	5.65	12.07	32.99
rank:	6	6	6	6	24

In het corpus van de tabel staan zogenaamde gestandaardiseerde afwijkingen tussen geobserveerde en verwachte frequenties. Deze afwijkingen kunnen bij

benadering geïnterpreteerd worden als standaardnormale afwijkingen. Doch let wel: deze interpretatie is niet helemaal correct, omdat de cellen onderling afhankelijk zijn, en er geen rekening gehouden wordt met het feit dat de parameters uit de data zijn geschat. De kolom helemaal rechts is de som der kwadraten van de rij-elementen, en kan bij benadering geïnterpreteerd worden als een chi-kwadraat variabele met aantal vrijheidsgraden gelijk aan aantal groepen minus één (hier dus 4-1=3). Onderaan de tabel verschijnt de som der kwadraten van de kolom-elementen. Het effect van het rekening houden met de schatting van de parameters komt tot uiting door vergelijking van de rij 'sum sq' en 'R1c'. Meestal (doch niet altijd) is de R1c-bijdrage kleiner dan de 'sum sq'. De afzonderlijke R1c-bijdragen kunnen bij benadering geïnterpreteerd worden als een chi-kwadraat-verdeelde grootheid met aantal vrijheidsgraden als aangegeven in de rij 'rank'. Dit is ook slechts een benadering, omdat er geen rekening is gehouden met het aantal geschatte parameters. Immers, de som van de R1c-bijdragen (32.99) heeft geen 24 maar slechts 18 vrijheidsgraden, omdat er zes vrije parameters uit de data zijn geschat. De rij R1c blijft achterwege in deze tabellen indien geanalyseerd is met de optie 'item tSts = D'.
De herkomst van de gestandaardiseerde afwijkingen kan gedeeltelijk achterhaald worden door inspectie van de 'huge' uitvoer:

HUGE OUTPUT

Tables for R1c-statistic
==========

Booklet 1
==========

booklet 1, group 1; score-range: 1 - 2; number of obs. = 257

item	label	cat.	obs.	exp.	dev.	scaled dev.
1	tv= -1.5	1	147	146.16	.84	.11
2	tv= -1.0	1	109	105.09	3.91	.51
3	tv= -0.5	1	76	68.79	7.21	1.04
4	tv= 0.0	1	32	46.87	-14.87	-2.44
5	tv= 0.5	1	29	23.64	5.36	1.16
6	tv= 1.0	1	19	18.60	.40	.10
7	tv= 1.5	1	7	9.85	-2.85	-.93

sum of squares =	9.53
contribution to R1c =	8.33
rank of matrix of weights =	6

Het aantal tabellen voor de R1-toets in de 'extended' uitvoer is gelijk aan het aantal boekjes. In de 'huge' uitvoer zijn er per boekje evenveel tabellen als er groepen zijn. Indien de steekproef per boekje niet te klein is, zijn dat er meestal vier. Bij tien boekjes kan men dus veertig tabellen krijgen. Het aantal rijen in de tabel is $\Sigma_i m_i$, waarin i loopt over alle items in het boekje, en m_i de hoogste score is die men op het item kan halen.
De kolom dev. (deviation) is het verschil tussen de kolom 'obs' (dat is het geobserveerde aantal antwoorden in de categorie) en de kolom 'exp' (verwacht aantal). De kolom 'scaled dev.' is de gestandaardiseerde afwijking. Deze kolom is ook integraal terug te vinden in de R1c-tabellen van de 'EXTENDED' uitvoer.
Let wel: de gestandaardiseerde afwijkingen zijn niet eenvoudig uit de tabellen te berekenen. Hetzelfde geldt ook voor de X^2- en M-toetsingsgrootheden. Zelfs met de optie 'item tSts = D' kunnen de toetsingsgrootheden niet zonder meer uit de afgedrukte tabellen worden berekend. De reden hiervan is, dat de voorspelde proporties $\pi_{i|s}$ ingewikkelde functies van de (geschatte) item-parameters zijn.

6.2.2 Het programma OPLOOK

Het programma OPPRE leest de data van het databestand volgens de specificaties die door de gebruiker via OPIN zijn opgegeven. Alle informatie die de programma's OPCML en OPMML nodig hebben, worden in een speciaal formaat weggeschreven in het .PRE-bestand. Het programma OPLOOK doet niets anders dan dit .PRE-bestand lezen en de inhoud, voorzien van enig summier commentaar, wegschrijven naar het .LK-bestand. Dit bestand kan via een editor op het scherm geprojecteerd worden, of naar de printer worden gestuurd.
.PRE-bestanden kunnen heel erg groot worden. De oorzaak hiervan is, dat voor elk item in elk boekje en voor elke score moet worden weggeschreven hoeveel personen met die score het item juist hebben beantwoord. Deze getallen zijn conditionele frequenties: hoeveel personen hebben item i juist, gegeven dat hun score gelijk is aan s. Deze getallen zijn nodig om de geobserveerde frequenties te kunnen groeperen. (Ze kunnen niet vooraf gegroepeerd worden omdat de groepering afhangt van de itemparameters.) Er zijn op het Cito gegevens geanalyseerd waarvoor het .PRE-bestand groter was dan vier megabytes. Om zeer grote .LK-bestanden te vermijden, worden deze conditionele frequenties standaard niet naar het .LK-bestand geschreven. Ze

zijn echter wel op te vragen, door bij de aanroep van OPLOOK de schakelaar /CF mee te geven.

Voorbeeld. Hierna volgt een uittreksel uit het bestand EXAMPLE.LK. Het gaat om de analyse van 15 items aangeboden in twee boekjes: boekje 1 bevat de items 1 tot 10, boekje 2 de items 6 tot 15. De maximum score (maXs) op elk item is 2. De oneven items hebben een a-waarde van 1, de even items een a-waarde van 2. De maximale score voor elk boekje is dus 30. OPLOOK wordt opgeroepen met het commando:

OPLOOK [EXAMPLE] /CF [CR]

De conditionele frequenties worden weggeschreven als laatste informatie op het .LK-bestand. Hieronder staan de inleidende informatie voor deze frequenties en de conditionele frequenties voor item 6.

```
          Cumulative number of category responses given score
          (score range: 1 to maximum-1)

item   6 (categories ≥ 2) :
        booklet 1:   0   0   0   0   0   0   0   0   0   0
                     0   0   0   1   1   1   2   0   4   3
                     7   7   3   2   8  10   9  17   6
        booklet 2:   0   0   0   0   0   2   2   1   1   1
                     3   2   2   3   2   3   7   4   8   6
                     8   8   6   4   8   6   7   6   2

item   6 (categories ≥ 1) :
        booklet 1:   0   0   0   0   1   3   3   2   4   5
                     5   9   8   9  12  14   9   9  12  13
                    17  19  17  10  17  13  13  18   6
        booklet 2:   0   4   5  11  13  13  12  12  10  14
                    11  10  12  10   3   8  13   7  14  14
                    10  10  10   7   9   7   8   6   2
```

De getallen die afgedrukt worden, zijn de (cumulatieve) conditionele score-frequenties. De positie van het getal weerspiegelt de score: het eerste getal in elke rij is de conditionele frequentie gegeven dat de score = 1, het tweede idem voor score = 2, tot de maximum score minus 1. (Per rijtje worden er dus 29 getallen afgedrukt, 10 per regel). De frequenties zijn cumulatief over de categorieën: de onderstreepte '8' (score 21) voor boekje 2 betekent dus dat er 8 personen zijn met score 21 die op item 6 2 of meer

(dus 2) punten hebben behaald; de onderstreepte '10' betekent dat er 10 personen met score 21 1 of meer punten hebben behaald. Er zijn dus 10 - 8 = 2 personen die precies 1 punt hebben behaald.

6.3 Het programma OPSUG

In hoofdstuk 4 hebben we toegelicht dat de discriminatie-indices tijdens de analyse de status krijgen van een per hypothese ingevoerde constante. De itemgerichte X^2-toetsen en de M-toetsen zijn speciaal ontworpen om deze hypothesen te toetsen, en bovendien geven de M-toetsen en de grafische uitvoer van de X^2-toetsen aanwijzingen in welke richting deze indices het beste kunnen worden aangepast. Men moet echter de kracht van deze toetsen niet overschatten: de toetsen zijn wel gevoelig voor misspecificatie van een bepaalde discriminatie-index, doch niet uitsluitend hiervoor. In principe tast één verkeerde index alle toetsen aan. Als we er dus op grote schaal naast zitten met al die hypothesen kunnen we niet zoveel aanvangen met de itemgerichte toetsen. Het programma OPSUG doet aan de hand van een speciale analyse van de data een SUGgestie voor een redelijke hypothese. (Technisch: deze analyse is een soort kleinste-kwadraten-schatting van zowel item- als persoonsparameters in het Birnbaummodel. Het probleem wordt verder gereduceerd door aan te nemen dat ϑ in subgroepen met ongeveer dezelfde ongewogen score constant is. De schattingen met dit algoritme zijn zeker niet consistent, maar in de praktijk blijken ze erg nuttig te zijn als 'suggestie' voor de discriminatie-indices.)
De procedure die OPSUG volgt is echter niet volautomatisch. Er dient één belangrijke beslissing genomen te worden door de gebruiker. Het programma OPSUG maakt een schatting van de a-waarden in het domein van de positieve reële getallen, doch de a-waarden voor OPCML en OPMML dienen gehele getallen te zijn. Er moet dus afgerond worden. Maar in hoofdstuk 4 hebben we eveneens gezien dat de absolute waarde van de discriminatie-indices er niet toe doet, alleen hun onderlinge verhouding is belangrijk. Dit betekent dat we de fractionele a-waarden die OPSUG vindt, met een willekeurige positieve constante mogen vermenigvuldigen alvorens af te ronden. De onderlinge verhouding van de getransformeerde fractionele a-waarden

verandert niet, doch de verhouding na de afronding kan wel degelijk en drastisch veranderen, zoals blijkt uit Tabel 8.

Tabel 8

Effect van de transformatie op de afgeronde a-waarden

voor afronding				na afronding			
a_1	a_2	a_3	geom.gem.	a_1	a_2	a_3	geom. gem.
1.0	1.4	1.2	1.19	1	1	1	1.00
2.0	2.8	2.4	2.38	2	3	2	2.29
3.0	4.2	3.6	3.57	3	4	4	3.63
4.0	5.6	4.8	4.76	4	6	5	4.93

In de linkerhelft van de tabel staan voor drie items de a-waarden, per rij telkens vermenigvuldigd met een andere constante (resp. 1, 2, 3 en 4). In de rechterhelft staan de afgeronde waarden die geschikt zijn voor OPLM. De verhouding tussen de a-waarden rechts is niet precies gelijk aan de overeenkomstige verhouding links. (De verhoudingen zouden precies gelijk zijn als we met 5 vermenigvuldigden.) In het algemeen geldt dat, hoe groter de constante is waarmee vermenigvuldigd wordt, des te beter na afronding de oorspronkelijke verhoudingen benaderd worden. Echter, om die constante verstandig te kunnen kiezen, moeten we de grootte-orde kennen van de a-waarden die OPSUG berekent. In Tabel 8 wordt impliciet aangenomen dat de bovenste rij in de linkerhelft het resultaat is van OPSUG. In werkelijkheid liggen die grootte-orden helemaal anders, en in het algemeen zijn ze intuïtief niet goed aan te voelen. Om dit probleem te omzeilen, kunnen we de gemiddelde grootte van de a-waarden (voor afronding) zelf bepalen. Als maat voor de gemiddelde grootte is het geometrische gemiddelde gekozen. (Het geometrische gemiddelde van k positieve getallen is de k-de -machtswortel uit het product van die getallen; het is in het algemeen iets kleiner dan het rekenkundig gemiddelde.) De gebruiker dient bij aanroep van OPSUG dit gemiddelde op te geven. De a-waarden die OPSUG bepaald heeft, worden dan herschaald zodat ze precies het opgegeven geometrisch gemiddelde hebben. Daarna worden ze afgerond tot het dichtstbijgelegen gehele getal.

Het geometrische gemiddelde van de afgeronde a-waarden wordt meegedeeld in het .CML-bestand, direct onder de centrale tabel. De afrondingsregel kent twee uitzonderingen: als het afgeronde getal gelijk is aan nul, wordt het toch op 1 gezet, en als het groter is dan 15 wordt het afgerond op 15. (De minimale a-waarde in OPLM is 1, de maximale 15.)
Het kiezen van het geometrisch gemiddelde is geen gemakkelijke taak. Wordt het te klein gekozen, dan gaan door afronding belangrijke verschillen verloren, maar als het te groot gekozen wordt, dan bestaat het gevaar dat er gekapitaliseerd wordt op toeval: nuances die toevallig in de steekproef aanwezig zijn (maar in een volgende onafhankelijke steekproef niet meer), krijgen extra aandacht, terwijl ze 'in de populatie' niets voorstellen. Ervaring met OPLM heeft uitgewezen dat een geometrisch gemiddelde van ongeveer 3 in veel gevallen een goede waarde is om de analyses mee te beginnen. Er moet echter rekening gehouden worden met de volgende algemene kenmerken van de steekproef en de items:

1. Het programma deelt de respondenten die hetzelfde boekje gekregen hebben, in een aantal groepen in met ongeveer gelijke score. Om voldoende stabiliteit van de geschatte a-waarden te verkrijgen, mogen die groepen niet te klein zijn. Indien de steekproef (per boekje) zo klein is dat er maar één groep gevormd kan worden, is het niet mogelijk de verschillende discriminatie-waarden van de items te schatten, en krijgen alle items dezelfde a-waarde. De itemgerichte toetsen kunnen echter wel significant worden omdat de gegevens van één item gecombineerd worden voor alle boekjes waar dit item in voorkomt.

2. De stabiliteit van een geschatte a-waarde is afhankelijk van de p-waarde van het item. Bij extreme p-waarden (bijvoorbeeld groter dan .90) is die stabiliteit niet erg groot, en ook de gevoeligheid van de toetsen voor misspecificatie is niet al te groot. Bij een nieuwe analyse op een steekproef waarbij de p-waarde van hetzelfde item minder extreem is, kan blijken dat de eerder gekozen waarde niet goed gekozen was.

Het programma OPSUG maakt gebruik van het .SCR-bestand en van het data-bestand. De uitkomst van OPSUG is onafhankelijk van de discriminatie-

indices die aan OPIN zijn meegedeeld. Het programma maakt steeds het bestand JOBN.SUG aan, waarin een tabel staat met de afgeronde resultaten (zie Tabel 9). Optioneel kunnen de resultaten ook rechtstreeks in het .SCR-bestand geschreven worden. Het programma begint met een kleine dialoog waarin om deze optie en om het geometrische gemiddelde gevraagd wordt.

Tabel 9

Voorbeeld van een .SUG bestand

Suggestion of discrimination-indices for job EXAMPLE
with data read from EXAMPLE.DAT

nr	A	#b	nr	A	#b	nr	A	#b
1	1	1	6	2	2	11	1	1
2	2	1	7	1	2	12	2	1
3	1*	1	8	2	1	13	1	0
4	2	1	9	1	2	14	2	1
5	1	1	10	2	2	15	1	1

These results are not written on .\EXAMPLE.SCR

In het voorbeeld in Tabel 9 zijn de 15 items in twee boekjes afgenomen: boekje 1 bevat de items 1 tot 10, boekje 2 de items 6 tot 15. De items 6 tot 10 kwamen dus in beide boekjes voor. De suggestie '1*' voor item 3, betekent dat in het programma de afgeronde index gelijk was aan nul. De kolom '#b' duidt het aantal boekjes aan dat effectief gebruikt is om de index te schatten. In sommige gevallen kan het programma de gegevens uit een boekje niet gebruiken om de a-waarde te schatten. Deze gevallen komen grosso modo (doch niet exact) overeen met de items die een negatieve r_{it}-waarde hebben. Item 8 komt weliswaar in beide boekjes voor, doch de informatie in één van beide boekjes was onbruikbaar om a_8 te schatten. De index voor item 13 kon niet geschat worden (#b = 0). Het programma zet de index op zijn minimale waarde 1. De laatste regel van Tabel 9 weerspiegelt de keuze die de gebruiker in de dialoog heeft gemaakt.

6.4 Het programma OPREAD

Met dit programma kunnen parameterschattingen (met OPCML) als 'fiXed' waarden worden ingevoerd in OPIN. Het programma leest de parameterwaarden uit een .PAR-bestand en schrijft ze in een .SCR-bestand. Er kan een willekeurige keuze gemaakt worden uit de items in het .PAR-bestand en de waarden uit dit bestand kunnen naar willekeurige items in een willekeurig .SCR-bestand geschreven worden. Zo kan de parameterwaarde van item 1 uit een eerdere analyse (de oorsprong) als bekende constante worden ingevuld voor item 13 (de bestemming) in een volgende analyse. Dit kan zinvol zijn als item 13 in de tweede analyse hetzelfde item is als item 1 in de eerste analyse. De items die oorsprong en bestemming aangeven moeten wel dezelfde maximumscore hebben. Het programma wordt aangeroepen met het commando **OPREAD** [CR] en het begint met een dialoog. Hierna volgt een voorbeeld van zo'n dialoog. De antwoorden van de gebruiker zijn onderstreept, commentaar staat *cursief*.

.SCR-file [.SCR] : example *(getallen worden weggeschreven in EXAMPLE.SCR)*

.PAR-file [.PAR] : example *(getallen worden gelezen uit EXAMPLE.PAR)*

Itemnumbers on parfile example.PAR are
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

Select itemnumbers (ranges allowed):
(1)-->[CR] 1 2 3 [CR]
(2)-->[CR] 4-11 [CR] *(de items 1 tot 11 zijn geselecteerd)*
(3)-->[CR] [CR] *(itemnummers worden opgevraagd tot alleen [CR] of tot de lijst is uitgeput)*

Give corresponding numbers on .SCR file (idem=CR)
(1)-->[CR] [CR] *(oorsprong-nummer en bestemmingsnummer zijn identiek)*

Copy discrimination indices as well (Y/N) ? n

De oorspronkelijke reden waarom het programma OPREAD werd geschreven, was om snel kruisvalidaties te kunnen doen. Bij een kruisvalidatie worden gegevens verzameld op twee onafhankelijke steekproeven (of een steekproef wordt 'at random' in twee gelijke helften gesplitst). De itemparameters worden geschat uit de gegevens van de eerste steekproef, de schattingen

worden vervolgens als constanten ingevoerd, en de gegevens van de tweede steekproef worden geanalyseerd. De redenering achter deze procedure is, dat als het model goed past bij de eerste steekproef en het heeft populatie-geldigheid, dan moet het ook goed passen bij de tweede steekproef. Deze redenering is echter niet helemaal juist: de parameterschattingen bij de eerste steekproef zijn behept met een schattingsfout. Omdat de parameters de gegevens zo goed mogelijk proberen te beschrijven, weerspiegelen de schattingsfouten als het ware de specifieke eigenaardigheden van de eerste steekproef. Vervolgens worden de schattingen als populatiewaarden beschouwd om de gegevens van de tweede steekproef te analyseren. Een even goed resultaat (met de statistische toetsen) impliceert dan eigenlijk de verwachting dat de eigenaardigheden van de tweede steekproef gelijk zijn aan die van de eerste steekproef en dit is niet realistisch want de twee steekproeven worden geacht onafhankelijk van elkaar te zijn. Men dringt als het ware de schattingsfouten van de eerste analyse op aan de tweede steekproef. Daardoor zal de passing bij kruisvalidatie in het algemeen slechter zijn dan bij de oorspronkelijke analyse. Het is wel zo dat de invloed van de schattingsfouten kleiner wordt naarmate beide steekproeven groter worden, maar precieze kwantificaties zijn niet bekend. De bruikbaarheid van OPREAD is dus erg beperkt.

6.5 Het programma OPSIM

Het programma OPSIM is een volledig op zichzelf staand programma waarmee artificiële gegevens kunnen worden aangemaakt volgens OPLM. Het commando om het programma te beginnen is:

OPSIM [CR]

Het programma opent dan een dialoog om de toetsspecificaties te maken. Hierna volgt deze dialoog, met voorbeeld-antwoorden (onderstreept) en commentaar (*cursief*).

Number of items ? 10 [CR] *(maximum aantal items is 80)*

Discrimination parameters (10 numbers) ? 10*1 [CR]
*discriminatieparameters mogen fractionele getallen zijn. 4*1 is hetzelfde als 1 1 1 1. Indien er te weinig getallen zijn opgegeven*

wacht het programma op de volgende regel op de rest (zonder prompt !). Indien er teveel getallen zijn opgegeven, wordt het teveel niet gelezen.

Standard deviation of population: 2 [CR]

De θ-waarden worden getrokken uit een normale verdeling met een gemiddelde van nul en een standaard afwijking als hier aangegeven. Zie echter ook het commentaar na dit voorbeeld.

Maximum score per item (10 numbers) ? 10*1 [CR]

Category parameters (10 numbers) ? 3*-1 4*0 3*1 [CR]

Bij dichotome items worden hier de itemparameters opgegeven. Bij polytome items de categorieparameters; eerst alle parameters voor het eerste item, dan voor het tweede enz.

Ranknumber of test (booklet) ? 1 [CR]

De antwoorden en het boekjesnummer dat hier wordt opgegeven worden naar het uitvoerbestand geschreven. Het boekjesnummer komt in kolom 1, de antwoorden beginnen in kolom 2, één kolom per antwoord. Het boekjesnummer mag dus niet groter zijn dan 9.

Number of observations ? 1000 [CR] *(aantal moet kleiner zijn dan 32768)*

Output file ? DATA [CR]

Seed of random sequence determined by program ? (Y/N) Y [CR]

By 'Y' kiest het programma zelf een beginwaarde, gebaseerd op de klok van de machine. Deze waarde wordt op het scherm geschreven. Eenzelfde specificatie van de toets gecombineerd met dezelfde startwaarde geeft exact dezelfde resultaten.
Bij 'N' moet de gebruiker zelf een startwaarde opgeven. Deze startwaarde moet een geheel getal zijn tussen 1 en 2,147,483,647 (= 2^{31} - 1). Dit getal moet bij voorkeur groot en oneven zijn.

Seed generated by the program is 538902527

Analyse van de aldus gegenereerde data zal op de schattingsfout en een verschuiving na, de parameters opleveren die in OPSIM zijn gespecificeerd. OPCML normeert de schattingen zodanig dat de som van de parameters nul is. Indien de som van de werkelijke parameters die men bij OPSIM opgeeft niet nul is, zal er een verschuiving plaatsvinden. OPMML geeft twee verschillende normeringen: één als in OPCML en één met het gemiddelde gelijk aan nul.

Het voorgaande geldt natuurlijk slechts indien in OPIN dezelfde discriminatie-indices zijn opgegeven als die welke gebruikt werden voor het genereren van de data.
OPSIM kan ook gebruikt worden om bestanden te maken met onvolledige designs. Stel, men wil een bestand maken, waarbij het eerste boekje de antwoorden bevat op de items 1 tot 10, en het tweede boekje die op de items 6 tot 15. Door twee keer OPSIM te draaien (met een verschillend 'seed' voor de random-getallen-generator) en er voor te zorgen dat de laatste vijf parameters van de eerste run gelijk zijn aan de eerste vijf parameters van de tweede run, worden de gegevens voor de twee boekjes weggeschreven in twee verschillende bestanden (bijvoorbeeld DATA1 en DATA2). Denk erom ook het boekjesnummer aan te passen. Door de eenvoudige DOS-instructie

Copy DATA1+DATA2 DATA [CR]

bevat het bestand DATA alle gegevens die direct met de optie 'Structure = Compressed' geanalyseerd kunnen worden.

7 OPLM met polytome items

7.1 Het polytome OPLM-model

Bij een dichotoom item worden twee antwoordcategorieën onderscheiden: 0 en 1. Deze categorieën worden als geordende categorieën geïnterpreteerd: een juist antwoord (1) wijst op een hogere vaardigheid dan een verkeerd antwoord (0). Bij polytome items neemt men aan dat de score op item i $m_i + 1$ verschillende waarden kan aannemen (0, 1, ..., m_i), waarbij een hogere itemscore wijst op een grotere vaardigheid dan een lagere itemscore. De itemresponsfunctie bij dichotome items is de kans dat het item juist wordt beantwoord; deze functie is een monotoon stijgende functie van ϑ. Deze functie beschrijft de kans dat de antwoordvariabele $X_i = 1$; strikt genomen kunnen we zeggen dat de itemresponsfunctie bij een dichotoom item de categorieresponsfunctie van categorie 1 is. We hadden natuurlijk evengoed de categorieresponsfunctie van categorie 0 kunnen kiezen als itemresponsfunctie. Omdat $\text{Prob}(X_i = 0|\vartheta) = 1 - \text{Prob}(X_i = 1|\vartheta)$ kunnen we volstaan met één van beide functies; de andere ligt automatisch vast. In het polytome geval is een item gekenmerkt door $m_i + 1$ categorieresponsfuncties; m_i is de maximum score die op het item kan behaald worden. Stel bijvoorbeeld $m_i = 2$, dan kunnen we de kans beschouwen dat de antwoordvariabele X_i de waarde 0 aanneemt als functie van ϑ. Deze functie is de categorieresponsfunctie voor categorie 0. Voor de waarden 1 en 2 bekomen we dan de categorieresponsfuncties voor categorieën 1 en 2. Omdat voor elke ϑ geldt dat

$$\text{Prob}(X_i = 0|\vartheta) + \text{Prob}(X_i = 1|\vartheta) + \text{Prob}(X_i = 2|\vartheta) = 1$$

liggen de drie functies volledig vast als we er twee kennen. In het algemeen: als m_i functies gegeven zijn, liggen alle $m_i + 1$ functies vast. De functieregel voor het polytome model is gegeven door

$$\text{Prob}(X_i = j \mid \hat{\vartheta}) = \frac{\exp[a_i(j\hat{\vartheta} - \sum_{g=1}^{j} \beta_{ig})]}{1 + \sum_{h=1}^{m_i} \exp[a_i(h\hat{\vartheta} - \sum_{g=1}^{h} \beta_{ig})]}, \qquad (j = 1, \ldots, m_i), \qquad (34a)$$

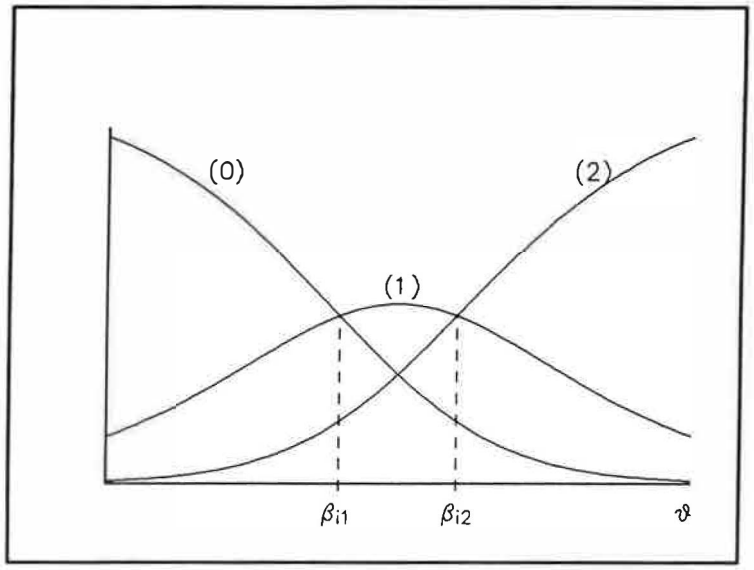
en door het feit dat de probabiliteiten optellen tot 1, volgt noodzakelijkerwijs dat

$$\text{Prob}(X_i = 0 \mid \hat{\vartheta}) = \frac{1}{1 + \sum_{h=1}^{m_i} \exp[a_i(h\hat{\vartheta} - \sum_{g=1}^{h} \beta_{ig})]}. \qquad (34b)$$

De grootheid a_i is als vanouds de discriminatie-index van item i. Als we m_i gelijkstellen aan 1 zien we dat formules (34a) en (34b) formeel gelijk zijn aan formules (21a) en (21b) met $\sigma_i = \beta_{i1}$: de formules (34) zijn dus ook geldig voor het dichotome geval, met dien verstande dat wat we in het dichotome geval de 'moeilijkheidsparameter' van het item genoemd hebben, nu iets algemener de categorieparameter van categorie 1 van item i genoemd wordt. (De eerste index van β verwijst naar het item, de tweede naar de categorie.) De interpretatie van de categorieparameters is het duidelijkst aan de hand van de grafiek van (34). In Figuur 21 zijn de drie functies afgebeeld voor item i met $m_i = 2$, $a_i = 1$, $\beta_{i1} = -0.5$ en $\beta_{i2} = +0.5$. β_{i1} is het punt op de ϑ-as waar de curves voor de categorieën 0 en 1 elkaar snijden: voor die waarde van ϑ zijn de kansen op een 0- en een 1-antwoord even groot; voor β_{i2} geldt analoog dat de kansen op een 1- en een 2-antwoord even groot zijn. In het algemeen geldt: β_{ij} is de waarde van ϑ waarvoor de kans op categorie j en j-1 even groot zijn. Bij Figuur 21 kunnen we verder nog het volgende opmerken:

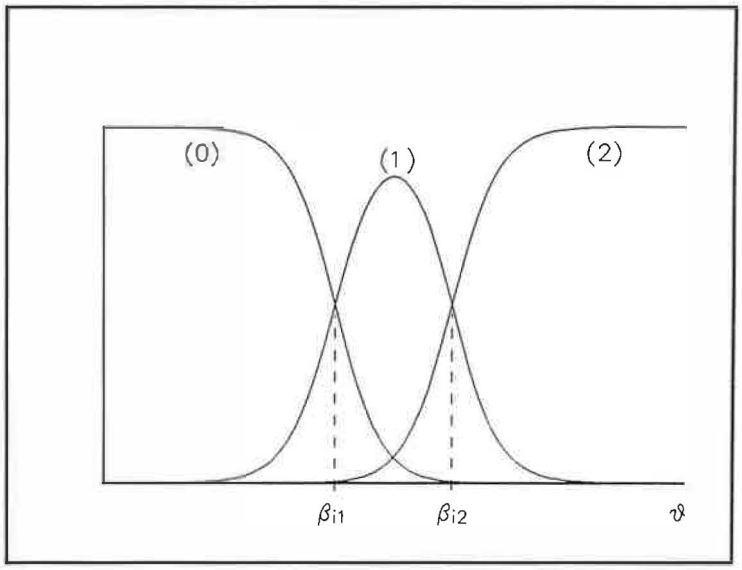
1 Voor zeer kleine ϑ-waarden is de kans op categorie 0 bijna 1: een zeer lage vaardigheid levert bijna zeker 0 punten op. We zien dat de kans op de 0-categorie monotoon afneemt als de vaardigheid stijgt. Omgekeerd geldt dat voor zeer hoge waarden van ϑ de kans op de hoogste categorie bijna 1 is, en in het algemeen dat de functie voor de hoogste categorie monotoon stijgend is. Deze twee kenmerken brengen automatisch met zich mee dat de curves voor de tussenliggende categorieën eerst stijgen en

daarna weer dalen. Dit kan niet anders, omdat de som van de curves altijd gelijk moet zijn aan 1.



Figuur 21

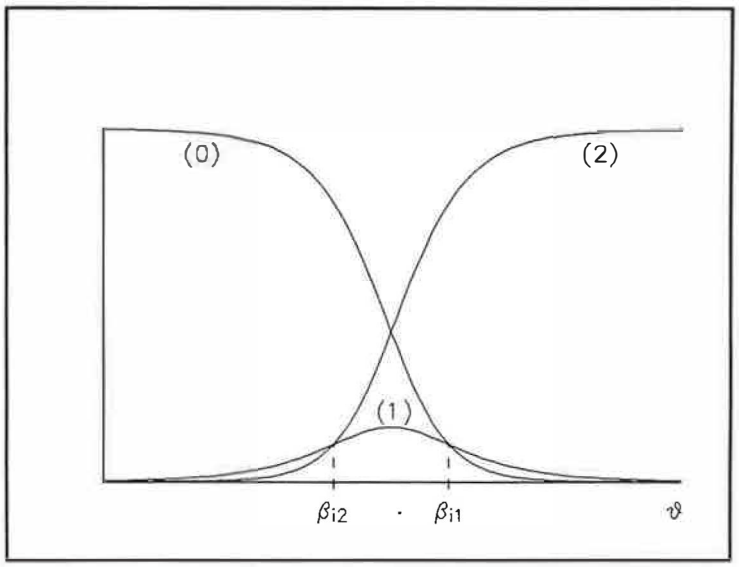
Categorieresponsfuncties met
$m_i = 2$, $a_i = 1$ en $\beta_{i1} < \beta_{i2}$



Figuur 22

Categorieresponsfuncties met
$m_i=2$, $a_i=5$ en $\beta_{i1}<\beta_{i2}$

2 Het effect van de toename van de discriminatie-index is dat de curves allemaal een steiler verloop kennen. Dit is goed te zien door vergelijking van de Figuren 21 en 22, die op dezelfde schaal getekend zijn. In Figuur 22 zien we dat het item erg goed discrimineert op de punten β_{i1} en β_{i2}: een ϑ-waarde die een beetje kleiner is dan β_{i1} resulteert met grote waarschijnlijkheid in de 0-categorie, terwijl een ϑ-waarde die iets groter is dan β_{i1} met grote waarschijnlijkheid in de 1-categorie zal eindigen. Mutatis mutandis geldt dezelfde redenering voor β_{i2}.



Figuur 23

Categorieresponsfuncties met
$m_i=2$, $a_i=2$ en $\beta_{i1}>\beta_{i2}$

3 Zowel in Figuur 21 als in Figuur 22 zien we dat links van β_{i1} de 0-categorie de meest waarschijnlijke of modale categorie is; tussen β_{i1} en β_{i2} is categorie 1 de modale categorie, terwijl rechts van β_{i2} categorie 2 modaal is. Deze eigenschap is echter niet algemeen: indien $\beta_{i1} > \beta_{i2}$, zoals in Figuur 23, dan is categorie 1 nooit de modale categorie. In Figuur 23 is duidelijk te zien dat een antwoord in categorie 1 voor geen enkele waarde van ϑ erg waarschijnlijk is. Het is een wijdverbreid misverstand te denken dat de β-parameters (of hun

schattingen) oplopende waarden moeten hebben. Het is heel goed mogelijk dat aan het model voldaan is en dat toch $\beta_{i1} > \beta_{i2}$. Als bij een item het correctievoorschrift wordt gegeven, nul punten toe te kennen voor een duidelijk fout antwoord en twee punten voor een duidelijk correct antwoord, en bij hoge uitzondering één punt voor een welomschreven twijfelachtig antwoord, dan volgt uit het correctievoorschrift zelf dat categorie 1 nooit een grote kans van optreden zal hebben. Toch kan dit een goed item zijn, als aannemelijk te maken valt dat het twijfelachtige antwoord op grotere vaardigheid wijst dan het duidelijk foute antwoord.

4 De categorieparameters kunnen niet zonder meer geïnterpreteerd worden als moeilijkheidsparameters van het item of van onderdelen van het item. Men hoort wel eens een 'stapjes'-interpretatie van polytome items. Stel dat een rekenitem i luidt '(7+5)/2 =', dan zou men kunnen redeneren dat het item uit twee stappen bestaat, namelijk de optelling en de deling, en bijvoorbeeld het correctievoorschrift zo maken dat een correcte optelling één punt oplevert en dat twee punten alleen gegeven worden indien zowel optelling als deling correct zijn. Hoewel op dit correctievoorschrift als zodanig niets valt aan te merken, is de interpretatie dat β_{i1} de moeilijkheidsparameter is van de optelling en β_{i2} de moeilijkheidsparameter van de deling, onjuist. Stel dat we ook een rekenitem j hebben dat luidt '(7+5)x2 =', dan zou uit de stapjesinterpretatie moeten volgen dat $\beta_{i1} = \beta_{j1}$, want het eerste stapje (de optelling) in beide items is identiek. Er is echter niets in het model dat deze uitspraak rechtvaardigt, en het kan aangetoond worden dat de waarde van de eerste parameter mee beïnvloed wordt door de relatieve moeilijkheid van het tweede stapje. De β-parameters van een item moeten dus gezamenlijk als item-kenmerken geïnterpreteerd worden, en de interpretatie is in wezen niets meer dan wat als commentaar bij de Figuren 21, 22 en 23 hierboven gegeven is. Dit hoeft niet per se als een nadeel beschouwd te worden: ook items waarvan de 'stapjes'-opbouw niet duidelijk is, of duidelijk niet aanwezig is (zoals bijvoorbeeld in de globale beoordeling van een essay op een vijf-puntsschaal) komen in aanmerking om met het polytome OPLM geanalyseerd te worden.
De enige criteria waarover we beschikken om de adequaatheid van het model te beoordelen zijn de statistische toetsen. Dit betekent dat het

criterium empirisch is; de overeenkomst tussen de antwoorden van de respondenten en de eigenschappen van het model bepalen het al dan niet geschikt zijn van het model.

5 In de literatuur is een speciaal geval van het polytome OPLM bekend en uitvoerig bestudeerd. Het betreft het geval waarin alle discriminatie-indices aan elkaar gelijk zijn, zodat we een natuurlijke uitbreiding krijgen van het Raschmodel tot polytome items. Dit model staat in de literatuur bekend als het 'partial credit model' (PCM). De auteur ervan is G. Masters die het model onder die naam in 1982 in Psychometrika introduceerde. Hetzelfde model - hoewel in een iets andere algebraïsche gedaante - was echter reeds in 1973 door E.B. Andersen in Psychometrika gepubliceerd.

7.2 Parameterschatting in het polytome OPLM

Net als bij het dichotome geval kunnen de parameters geschat worden met een CML-procedure en met een MML-procedure. Bij MML wordt de bijkomende veronderstelling gemaakt dat de respondenten aselect uit één of meer normale verdelingen getrokken zijn. Om CML te kunnen toepassen, dienen we voor iedere respondent zijn toetsscore te berekenen. De toetsscore is per definitie de som van de itemscores, elk vermenigvuldigd met de discriminatie-index van het item. Bevat een boekje bijvoorbeeld 10 items, vijf met discriminatie-index 1 en vijf met discriminatie-index 2, en kunnen op elk item maximaal 2 punten gehaald worden, dan is de maximum toetsscore 30. Verder verloopt de schatting geheel analoog aan de schatting in het dichotome geval. De formules die er voor nodig zijn worden alleen wat complexer; we zullen ze niet behandelen.
Zoals reeds aangestipt bij de behandeling van het model voor dichotome items (hoofdstuk 4), is het bestaan van de schattingen niet altijd gegarandeerd. Een noodzakelijke (doch niet voldoende) voorwaarde voor het bestaan van schattingen bij dichotome items is dat elk item minstens éénmaal juist en minstens éénmaal fout moet zijn beantwoord. Voor polytome items geldt dat voor elk item in elke categorie minstens één antwoord geobserveerd moet zijn. Er zijn nog andere voorwaarden waaraan voldaan moet

zijn, die echter niet precies bekend zijn. Het programma controleert enkele voorwaarden en geeft een foutmelding indien er niet aan voldaan is. Indien schattingen niet bestaan of niet uniek zijn, maar dit wordt door het programma niet opgemerkt, krijgt men in de regel vreemde uitvoer: zeer grote standaardfouten of zeer grote waarden voor de itemgerichte X^2-toetsen. Dit is echter alleen het geval met de optie 'Est. proc = Complete'. Met de optie 'Diagonal' kan men een oplossing krijgen die niet uniek is, dat wil zeggen dat er oneindig veel oplossingen kunnen bestaan. Het programma vindt er één van, die puur afhankelijk is van de waarden waarmee de berekeningen begonnen zijn.
Het programma OPCML normeert zijn oplossing zo dat de som van alle β-parameters gelijk is aan nul. OPMML geeft twee oplossingen: één met dezelfde normering als OPCML en één met het gemiddelde van de (eerste) populatie gelijk aan nul.

7.3 Het toetsen van het polytome OPLM

7.3.1 De R_{1c}-toets

De R_{1c}-toets voor het polytome geval is een recht-toe-recht-aan uitbreiding van het dichotome geval: per boekje worden de respondenten ingedeeld in maximaal vier scoregroepen en binnen elke scoregroep worden voor elke categorie van elk item de geobserveerde en de verwachte frequenties met elkaar vergeleken. (De formule laten we achterwege.) Net als in het dichotome geval moet binnen één boekje dezelfde groepsindeling gehanteerd worden voor alle items, zodat voor sommige itemcategorieën gemakkelijk kleine verwachte frequenties kunnen ontstaan. Het gevaar hiervoor is groter bij polytome items dan bij dichotome items. Bij te veel kleine verwachte frequenties is de toets niet goed bruikbaar. Zie hierover hoofdstuk 5.

7.3.2 De itemgerichte X^2-toetsen en de M-toetsen

In hoofdstuk 4 is uiteengezet dat de itemgrichte X^2-toetsen en de M-toetsen vooral gevoelig zijn voor misspecificatie van de discriminatie-indices. Uit

Figuur 19 blijkt duidelijk dat bij dichotome items de afwijkingen tussen conditionele probabiliteiten $\pi_{i|s}$ en conditionele proporties $p_{i|s}$ een welbepaald patroon volgen. De wijze van groeperen en van berekenen van de toetsingsgrootheden is er speciaal op gericht dat dit soort afwijkingen elkaar niet of nauwelijks compenseren, zodat de toetsen gevoelig zijn voor misspecificatie van de a_i-waarden. Bij polytome items ligt de zaak iets ingewikkelder. Stel dat a_i onderschat is. De theoretische conditionele probabiliteiten $\pi_{ij|s}$ (dat is de probabiliteit dat categorie j is geobserveerd op item i gegeven dat de toetsscore gelijk is aan s) zullen een patroon volgen ongeveer gelijk aan de categorieresponscurven van Figuur 21. De geobserveerde proporties, die verlopen volgens de 'echte', grotere a_i zullen een verloop volgen ongeveer zoals in Figuur 22. Nu zou men kunnen proberen de twee figuren op elkaar te leggen, en te proberen in het algemeen de karakteristieke verschillen te beschrijven. Dit is niet eenvoudig, en er komt nog een complicerende factor bij: door de misspecificatie van de discriminatie-indices, zullen ook de β's die bij dit item horen systematisch verkeerd geschat worden, zodat de beide figuren niet simpelweg gesuperponeerd kunnen worden.
De oplossing die voor dit probleem bedacht is, is uiterst eenvoudig. Na de schatting van de parameters wordt het item gedichotomiseerd: als de hoogste itemscore gelijk is aan 3, worden bijvoorbeeld categorie 0 en 1 als een lage score beschouwd en categorie 2 en 3 als een hoge score. De conditionele probabiliteit op een hoge itemscore gegeven de toetsscore s, duiden we aan als $\pi^*_{i|s}$. In het voorbeeld is het duidelijk dat

$$\pi^*_{i|s} = \text{Prob}(X_i = 2|s) + \text{Prob}(X_i = 3|s) = \pi_{i2|s} + \pi_{i3|s}.$$

Een analoge definitie geldt voor de geobserveerde conditionele proporties. Met deze gedichotomiseerde items kunnen we dan precies dezelfde redenering toepassen als bij echte dichotome items, en de formules (31) en (32) waarin $\pi_{i|s}$ en $p_{i|s}$ vervangen worden door $\pi^*_{i|s}$ respectievelijk $p^*_{i|s}$ leveren de toetsingsgrootheden voor de X^2- en M-toetsen. De enige vraag die nog overblijft is waar we het item precies moeten dichotomiseren. Puur formeel maakt dit niets uit; de toetsingsgrootheden zijn chi-kwadraat verdeeld,

indien aan het model is voldaan. Voor het onderscheidingsvermogen van de toetsen maakt het natuurlijk wel uit: indien we een item waarbij categorie 0 bijna niet voorkomt, dichotomiseren zodanig dat alleen categorie 0 de lage score voorstelt, en alle andere categorieën de hoge score, dan heeft natuurlijk bijna niemand de lage score behaald, en zijn we verplicht zeer grove groepsindelingen te maken, waardoor de toets aan onderscheidingsvermogen zal verliezen. Anderzijds is het zo dat, indien op verschillende plaatsen gedichotomiseerd wordt en het model overeen komt met de werkelijkheid, de toetsen niet vaker significant mogen zijn dan het toeval toelaat (100*α%). Verschillende dichotomiseringen kunnen dus opgevat worden als een soort replicaties van soortgelijke toetsen. (Vergelijk met bloeddruk-onderzoek: Als je bloeddruk in orde is, verwacht je geen extreme meetuitkomsten als je bloeddruk een aantal keren gemeten wordt.) Daarom worden in het programma OPCML de items m_i keren gedichotomiseerd. Bij de eerste dichotomisering wordt alleen categorie 0 als lage score beschouwd; bij de tweede de categorieën 0 en 1, en bij de laatste wordt alleen de hoogste categorie als hoge score beschouwd. De notatie die we daarvoor gebruiken is afgebeeld in Tabel 10.

Tabel 10

Notatie voor het aanduidingen van dichotomiseringen

notatie voluit	verkorte notatie
[0-0:1-3]	[:1]
[0-1:2-3]	[:2]
[0-2:3-3]	[:3]

De verkorte notatie wordt gebruikt in de centrale tabel van de standaard uitvoer in het .CML-bestand. (Bovendien is door plaatsgebrek de expliciete aanduiding van de eerste dichotomisering '[:1]' weggevallen.) In de 'extended' uitvoer is de dichotomisering voluit genoteerd.

Per item worden er dus m_i X^2-toetsen uitgevoerd. Voor elke toets wordt een nieuwe aangepaste groepering gemaakt. Het aantal groepen (en dus het aantal vrijheidsgraden) kan per dichotomisering verschillen. Het item wordt dus op

m_i manieren bekeken. Voor de beoordeling van de toetsen geldt niet een soort meerderheidsregel: als twee van de vijf toetsen duidelijk significant zijn is er waarschijnlijk iets niet in orde met dit item (tenzij er een fout van de eerste soort is gemaakt, en dat weten we natuurlijk nooit met zekerheid). Het feit dat drie van de vijf toetsen niet significant zijn doet daar niets vanaf.
Per dichotomisering worden eveneens drie M-toetsen berekend. Voor de M2- en de M3-toets is de groepsindeling natuurlijk dezelfde; voor de M-toets geldt de aangepaste regel dat een score s naar de L-groep gaat indien $\pi^*_{i|s} \leq 0.4$ en naar de H-groep indien $\pi^*_{i|s} \geq 0.6$.

8 Het programma OPDRAW

Het programma OPDRAW geeft grafische uitvoer op het scherm. Het laat de grafieken zien die overeenkomen met Figuur 18, waarbij - na groepering in scoregroepen - de overeenkomst tussen conditionele probabiliteiten en conditionele proporties getoond wordt. Alle informatie die het programma nodig heeft, staat in het .PAR-bestand. Het programma kan opgeroepen worden met het commando

OPDRAW [CR]

of, met een argument, door in te tikken

OPDRAW JOHN [CR]

In beide gevallen verschijnt een beeldmerk op het scherm, dat verdwijnt als men 'return' ([CR]) intikt. Indien er geen argument is meegegeven in het commando verschijnt een keuzemenu als afgebeeld in Figuur 24.

```
O P D R A W    Parameter File
Directory  C:\OPLM
PAR File   EXAMPLE.PAR
No File
Run
Quit                              (c) Copyright Cito, R&M, 1991
```

Figuur 24

Menu van OPDRAW

De betekenis van de keuzen is de volgende

Directory: De directory waar het .PAR-bestand zich bevindt. Indien dit veld niet wordt ingevuld wordt het bestand gezocht in de directory die op dat moment actief is.

PAR File: De naam van het .PAR-bestand. (De extensie .PAR hoeft niet ingetikt te worden.) Intikken van *.* geeft een lijst van alle .PAR bestanden in de gekozen directory, waaruit gekozen kan worden door de cursor op de naam van het gewenste bestand te plaatsen en [CR] te tikken.

No File: Deze keuze maakt het mogelijk de categorieresponsecurven te tekenen voor zelfgekozen parameters. Deze keuze brengt de gebruiker direct in een speciale editor (zie verder bij @T).

Run: Schrijft een kopie van de centrale uitvoertabel naar het scherm. Vanuit deze tabel kan het item gekozen worden waarvoor men grafische uitvoer wenst.

Quit: Terug naar DOS. Men kan OPDRAW vanuit een willekeurig punt ook verlaten door net zolang 'Esc' (escape) te drukken tot men terug in DOS is. Vanuit het initiele menu voert 'esc' direct terug naar DOS.

Indien er met het commando een argument is meegegeven verschijnt direct de centrale tabel op het scherm. Deze centrale tabel verschilt in twee opzichten van de centrale tabel die naar het .CML-bestand is weggeschreven. (1) De standaardfouten worden niet meegedeeld en (2) er is een extra kolom, genaamd '(Nstat)'. Deze kolom geeft aan, aan hoeveel statistische groepen elk item is aangeboden. Tabel 11 is een uittreksel uit deze centrale tabel voor een voorbeeld met polytome items (m_i=2 voor elk item) die in een onvolledig design zijn aangeboden aan een groep jongens en een groep meisjes. Deze twee groepen zijn in OPIN als statistische groepen gedefinieerd. Item 5 is alleen aan de jongens aangeboden, item 6 zowel aan jongens als aan meisjes.

Tabel 11

Uittreksel uit de centrale tabel van OPDRAW

ParameterFile : C:\OPLM\EXAMPLE.PAR

ItNr	Label	Disc	Par(s)	ChiSqr	DF(NStat)	Prob	M	M2	M3
5	Arith	5	1 -1.525	3.436	5(1)	0.633	-0.344	-0.897	-0.955
		[:2]	-0.221	6.116	6(1)	0.410	1.258	0.848	1.352
6	Arith	6	2 -0.284	6.387	10(2)	0.782	2.148	1.209	1.198
		[:2]	0.713	3.031	6(2)	0.805	-0.044	0.321	0.169

In deze centrale tabel kan een item uitgekozen worden door de cursorbalk met de pijltjestoetsen op het item te plaatsen, of door F9 te tikken en vervolgens het itemnummer te kiezen. Per item zijn drie soorten grafische uitvoer beschikbaar: de categorieresponsfuncties, de informatiefunctie van het item en een grafiek van de conditionele probabiliteiten en proporties.

Deze worden zichtbaar door respectievelijk in te tikken @U, @I en @O (i.e. de letter o, niet het cijfer nul. Met het symbool '@' wordt de 'Alt'-toets bedoeld.)

@U: de categorieresponsfuncties. Na intikken van deze opdracht verschijnt een scherm met de categorieresponsfunctie van categorie 0. Achtereenvolgens aantikken van de spatiebalk laat de curven voor de volgende categorieën verschijnen. Indien $m_i>1$, verschijnt na de laatste functie de boodschap '@R: Regression' De regressiefunctie geeft de verwachte score op het item als functie van ϑ. (Deze verwachte score wordt nog eens gedeeld door de maximum score m_i.) Bij dichotome items valt de regressiefunctie samen met de item respons functie. De regressiefunctie is gegeven door

$$E(X_i|\vartheta) = \sum_{j=0}^{m_i} j\, Prob(X_i=j|\vartheta).$$

Deze regressiefunctie kan gebruikt worden om met één getal de moeilijkheid van het item uit te drukken: de moeilijkheidsgraad zou zinvol gedefinieerd kunnen worden als de waarde van ϑ waarvoor de regressiefunctie de waarde 0.5 aanneemt, d.i. de vaardigheid die gemiddeld de helft van het maximaal aantal te verdienen punten oplevert.

@I: de informatiefunctie. De informatiefunctie (als functie van ϑ) geeft aan hoeveel informatie een itemantwoord gemiddeld oplevert als het item wordt voorgelegd aan een persoon met die ϑ-waarde. (De precieze definitie wordt achterwege gelaten). Indien de informatiefunctie een kleine waarde aanneemt in een bepaald gebied van ϑ-waarden betekent dit dat het item voor dit gebied eigenlijk niet geschikt is: het antwoord kon van te voren al met grote kans op juistheid voorspeld worden. In het algemeen is het moeilijk om aan de hand van de informatiefunctie te beoordelen of een item al dan niet geschikt is om in een toets te worden opgenomen. De informatiefunctie is bij gebruik van OPLM dan ook van beperkte waarde. Voor toetsconstructie wordt verwezen naar het programma OTD.

@O: afwijkingen tussen observaties en voorspellingen. De volgende regels gelden voor de afbeeldingen:

1 Indien NSTAT=1 voor het gekozen item, verschijnen de grafieken voor alle dichotomiseringen tegelijk op het scherm. (zie Figuur 25).

2 Indien NSTAT>1, verschijnen NSTAT grafieken voor de eerste dichotomisering op het scherm. Andere dichotomiseringen verkrijgt men door het desbetreffende cijfer in te tikken: een 2 geeft de NSTAT grafieken voor de tweede dichotomisering ([:2]), enz.

De verwachte proporties per groep worden weergegeven door '~'tekens (in Figuur 25 verbonden); de geobserveerde proporties worden aangegeven door 'x' (in de figuur niet verbonden). De onderste en bovenste lijntjes geven de '95% betrouwbaarheids-omhullende' aan. Deze omhullende is slechts bij benadering juist, en in de regel iets te breed: het is goed mogelijk dat alle geobserveerde punten binnen de omhullende vallen, maar het resultaat van de X^2-toets toch significant is. De betrouwbaarheidsintervallen zijn asymmetrisch en zodanig dat het 95%-betrouwbaarheidsinterval steeds binnen het (0-1)-interval valt. (Technisch: het (asymptotische) betrouwbaarheidsinterval van de logit van de gemiddelde proportie voor de groepen wordt uitgerekend en met de inverse logit-transformatie teruggebracht naar het (0,1)-interval. Er worden twee soorten fouten gemaakt bij deze benadering: er wordt gerekend alsof de gemiddelde proportie binomiaal verdeeld is, terwijl ze samengesteld binomiaal verdeeld is, en er wordt geen rekening gehouden met de afhankelijkheid van de proporties in de verschillende groepen. De asymptotische variantie van de logit van een proportie p (met parameterwaarde π) is gegeven door $[n\pi(1-\pi)]^{-1}$.)

De verbindingslijntjes die in Figuur 25 gegeven worden, kunnen veranderd worden door de gebruiker. Daartoe dient het bestand OPINTFC.DEF te worden aangepast. Dit bestand bevat ook commentaar waarin staat hoe de veranderingen kunnen worden aangebracht. Bemerk echter wel dat de verbindingslijntjes tussen de theoretische probabiliteiten en de verbindingen tussen de betrouwbaarheidsgrenzen kromlijnig geïnterpoleerd worden, zodat ze steeds een vloeiend verloop hebben. De verbindingslijntjes tussen de 'x'-en zijn recht. Gaat men nu zowel de theoretische als de geobserveerde punten verbinden, dan is het goed mogelijk dat de rechte lijnen bijna volledig buiten de betrouwbaarheidsomhullende vallen, ook als de echte geobserveerde waarden (de 'x'-en zelf) zeer goed aansluiten bij hun verwachte waarde.

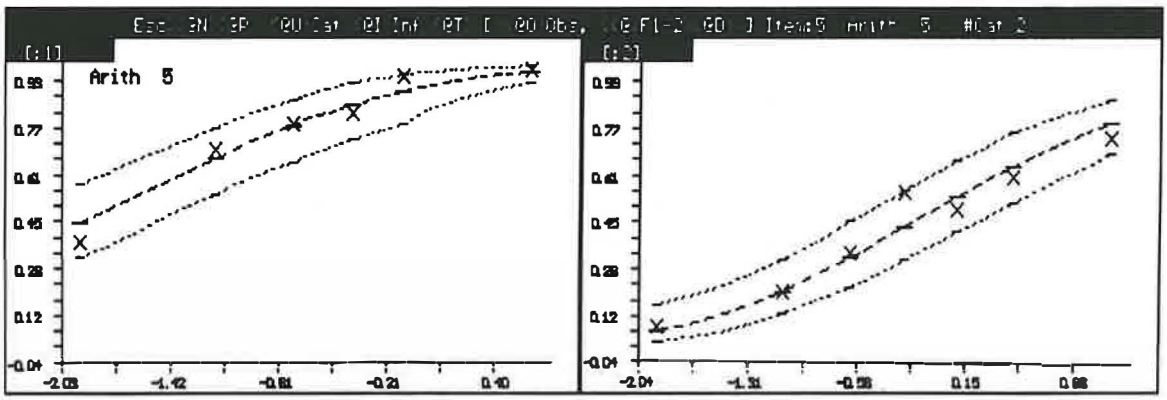
'esc' brengt de gebruiker terug naar de centrale tabel. Het is echter ook mogelijk de grafieken sequentieel te bekijken, zonder terug te gaan naar de centrale tabel:

CR: (next) geeft de volgende grafiek te zien. Dit is ofwel de eerste grafiek van het volgende item, ofwel de volgende dichotomisering van hetzelfde item (bij Nstat>1). Item 1 volgt op het laatste item.

●P: (previous) geeft de voorgaande grafiek te zien. Het laatste item gaat vooraf aan het eerste.

●D: (standardized deviates) Laat de afwijkingen tussen probabiliteiten en proporties op een ander wijze zien (zie Figuur 26). De nullijn komt overeen met de theoretische probabiliteiten, de proporties worden aangegeven door een speciaal symbool, al dan niet verbonden door rechte lijnen (te regelen via OPINTFC.DEF). Er wordt één plaatje per dichotomisering gemaakt. De gestandaardiseerde afwijkingen voor de verschillende statistische groepen worden in hetzelfde plaatje afgebeeld. De gestandaardiseerde afwijking in een groep van n observaties met proportie p en theoretische probabiliteit π is gegeven door

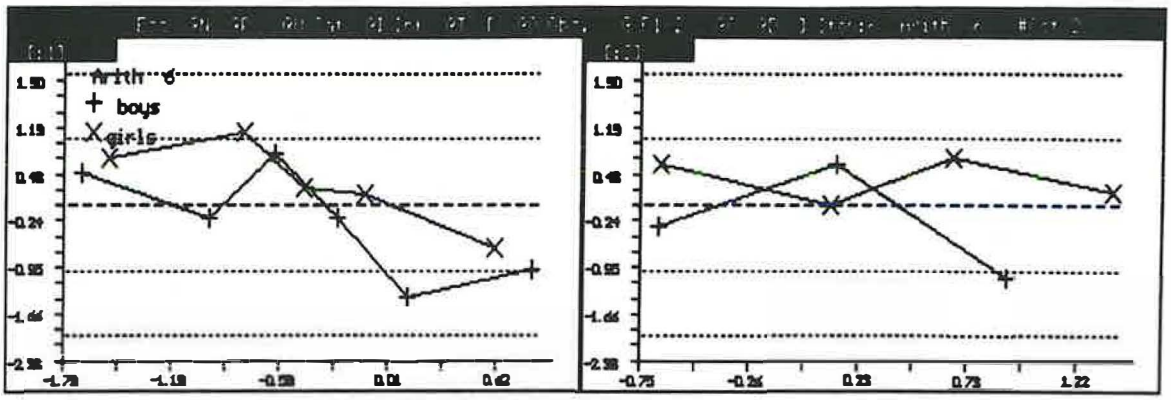
$$z = \frac{\sqrt{n}(p-\pi)}{\sqrt{\pi(1-\pi)}}.$$



Figuur 25

Afwijkingen tussen geobserveerde en voorspelde proporties voor item 5

De ordinaatwaarde van elk afgebeeld punt is dus het aantal z-waarden dat de geobserveerde proportie afwijkt van de overeenkomstige theoretische probabiliteit. Horizontale lijnen zijn getrokken op afwijkingen van ±1 en ±2. (Technisch: standaardiseren via de logittransformatie levert het probleem op dat geobserveerde proporties van 1 of 0 moeten afgebeeld worden op ± oneindig; daarom is deze transformatie hier niet gebruikt. Het kan dus voorkomen dat de positie van een proportie ten opzichte van de 95% betrouwbaarheidsgrens verschilt bij de twee voorstellingsmodi.)



Figuur 26

Gestandaardiseerde afwijkingen voor twee dichotomiseringen en twee statistische groepen

@D is een zogenaamde 'toggle': bij opnieuw intikken van @D verschijnen weer de niet-gestandaardiseerde afwijkingen.

@C: (Alleen bij niet-gestandaardiseerde afwijkingen en bij NSTAT>1) De default afbeelding van de afwijkingen gebruikt in elk plaatje een schaling van de ϑ-as zodanig dat de volledige breedte van het plaatje voor de grafiek benut wordt. Door @C krijgt men afbeeldingen met dezelfde schaling van de ϑ-as in elk plaatje. (@C is een 'toggle'.)

@F1, @F2...
Met deze toetsen kan men elk van de plaatjes die te zamen op één scherm verschijnen, uitvergroten tot het hele scherm. Het eerste plaatje met @F1, het tweede met @F2, enz. Het oorspronkelijke scherm wordt hersteld met @O.

@T: Met deze optie komt de gebruiker in een een-lijns-editor terecht, waar de discriminatie-index en de categorieparameters kunnen worden opgegeven. (Men komt eveneens in deze editor terecht met de keuze No File uit het initiele menu). Op de stippellijn kunnen maximaal tien getallen worden ingetikt; het eerste getal wordt geïnterpreteerd als de discriminatie-index (negatieve getallen toegelaten; nul wordt niet geaccepteerd), de volgende getallen worden geïnterpreteerd als de opeenvolgende β-parameters. Met @I of @U wordt een grafiek gemaakt van de informatiefunctie resp. de categorieresponsfuncties. Indien er een .PAR-bestand is ingelezen geeft @T op de editorlijn de parameters van het item waarop de cursorbalk staat in de centrale tabel. (Zie ook bij @Y.)

@Y: Bij de optie 'No File' identiek aan @T; Indien er een .PAR-bestand aanwezig is, komt @Y terug naar de een-lijns-editor met de waarden die er het laatst zijn ingebracht. (Zie ook bij @T.)

De huidige versie van OPLM voorziet geen mogelijkheden om de plaatjes op een eenvoudige manier naar papier te sturen. Deze optie is gepland voor een volgende versie van het programmapakket. Voor gebruikers van WordPerfect is het gebruik van de 'grab utility' wellicht nuttig. Het programma GRAB.COM bevindt zich in de WP-directory. Door het commando GRAB [CR] te geven wordt het programma GRAB resident, en kan het geactiveerd worden door de toetscombinatie Alt-Shift-F9. Slaat men deze combinatie aan terwijl een plaatje op het scherm staat, dan verschijnt een raampje op het scherm. Dit raampje kan verplaatst worden (met de cursorpijltjes) of vergroot en verkleind worden met Shift-cursorpijltjes. Geeft men vervolgens [CR] dan wordt de inhoud van het scherm binnen het raampje (grafisch) weggeschreven naar een .WPG-bestand dat opgeslagen wordt in de directory die op dat moment actief is. De naam van het bestand is GRAB.WPG voor het eerste, GRAB1.WPG voor het tweede enz. Deze bestanden kunnen in een WP document worden binnengehaald als figuur (Alt-F9,1).

Reeds eerder verschenen OPD Memoranda

92-1 P. Goldebeld. Toets- en Itemanalyse met TIA: Toelichting bij het lezen en interpreteren van toets- en itemanalyses voor gesloten en/of open vragen.

92-2 A.P.J.M. Heuvelmans. Contructie en verwerking van vragenlijsten.