

Evaluatie van statistische programmatuur; in het bijzonder m.b.t. MDS

Ulbe Brouwer en Pierre Debets.*

Samenvatting:

Erkele richtlijnen worden gegeven hoe programma evaluatie in de praktijk kan worden uitgevoerd. De nadruk ligt vooral op de gebruikersaspecten van programmatuur. Enkele uitgewerkte voorbeelden uit een daadwerkelijke evaluatie van een 6-tal multidimensional scaling programma's worden gerapporteerd.

Keywords: programmatuurevaluatie, multidimensional scaling, statistische simulaties, missing data.

1) Inleiding

Dienstverleningscentra die consultatie geven op het gebied van statistische programmatuur, krijgen dikwijls vragen voorgelegd in de trant van:

- welk programma voor faktoranalyse is voor mij het beste te gebruiken?
- is KYST beter te gebruiken dan MINISSA?
- waarom is EFAP beter dan FACTOR?
- welk faktoranalyse programma levert faktorscores volgens de Anderson-Rubin methode?
- is de rekennauwkeurigheid van BMDP beter dan die van SPSS?
- geeft MINISSA ook Shepardplots?

Het zal wel duidelijk zijn dat er vooral binnen dienstverlenende centra een grote behoefte bestaat om beschikbare (statistische) programmatuur te evalueren.

Met name op het gebied van veel gebruikte statistische methoden zoals b.v. de faktoranalyse bestaan er vele programma's met evenzovele of nog meer verschillende FA methoden. Tot nu toe is aan een grondige vergelijking van programma's of pakketten binnen een statistisch gebied vrij weinig gedaan. Het bekende werk van Francis (1979) geeft meer een inventarisatie van pakketten en hun inhoud dan een grondige vergelijking van de geïmplementeerde methoden met al hun invoer- en uitvoer opties.

In dit stuk willen we nagaan hoe een daadwerkelijke evaluatie van programmatuur, bedoeld voor een bepaalde algemene statistische methode (zoals b.v. re-

* Technisch Centrum FSW, Universiteit van Amsterdam.

gressietechnieken, variantie analyse, multidimensional scaling, faktoranalyse of loglineaire modellen) in de praktijk zou kunnen worden uitgevoerd.

Aan de hand van informatie uit de MDS subgroep van de CSP (contact groep statistische programmatuur) van de WS, zullen praktijkvoorbeelden worden gegeven.

Het zij opgemerkt dat evaluatie van programmatuur (binnen Nederland) tot nu toe uitsluitend binnen de CSP* heeft plaatsgevonden. In het verleden hebben de CLUSTER groep (van Meurs, 1978) en de MDS groep (Brouwer, Debets, van Schuur, 1980) "evaluatie" rapporten laten verschijnen, waarin ze de groepswerkzaamheden uiteengezet hebben.

Deze werkzaamheden zijn echter slechts te omschrijven als een poging om tot evaluatie te komen. In het volgende zullen we hierop terugkomen.

2) Wat is programma-evaluatie?

Wat voor soort vragen moeten beantwoord worden bij een programma-evaluatie? Voor we hierop ingaan willen we eerst met nadruk stellen dat de hier te behandelen programma-evaluatie uitsluitend plaatsvindt op het "gebruikersniveau", d.w.z. voor gebruikers van programmatuur moeten de evaluatie uitkomsten relevante informatie bevatten. Vragen die verband houden met meer technische aspecten van programmatuur, zoals: is het programma in een algemene hogere taal geschreven; is het goed gestructureerd en machine-onafhankelijk?, enz., komen hier niet aan de orde (deze vragen zijn echter wel relevant m.b.t. aanschafbeleid en onderhoudsmogelijkheden; met de meer inhoudelijke methodologische vragen hebben ze weinig te maken). We willen nu in het vervolg een programma-evaluatie zien als het beantwoorden van de onderstaande vragen:

- 1) welke programma's bestaan er voor een bepaalde methode of techniek?
- 2) wat zijn de mogelijkheden van elk programma: wat doen ze?
- 3) hoe ziet de gebruikersdokumentatie eruit?
- 4) hoe gebruikersvriendelijk is een programma?
- 5) hoe werkt het programma in de praktijk?
- 6) worden modelassumpties vooraf gecontroleerd?
- 7) hoe worden ontbrekende gegevens "verwerkt"?

Zelfs nadat deze vragen afdoende zijn beantwoord, blijft het in sommige gevallen lastig een duidelijke uitspraak te doen over het beste programma voor

* Per 1 januari '82 gaat de CSP op in de sectie statistische programmatuur (SSP) van de VVS.

een wel gespecificeerde statistische techniek.

Men kan zich het beste voorstellen dat de uiteindelijke evaluatiebevindingen worden weergegeven in een vergelijkend onderzoeksrapport zoals b.v. de consumentengids laat zien m.b.t. onderzochte apparatuur. Een dergelijk verslag bevat doorgaans veel tabellen en overzichtsschema's.

Een definitief eindoordeel, in de zin van een voorkeurslijst, wordt meestal niet gegeven. Elke gebruiker van programmatuur (cq. apparatuur) moet in staat geacht worden m.b.v. evaluatietabellen zelf een akseptabele keuze te doen. In paragraaf 4 willen we nu de genoemde vragen nader uitwerken en zo nodig voorbeelden uit een praktisch evaluatie onderzoek vermelden.

Hiervoor nemen we de bevindingen van de MDS subgroep van de CSP m.b.t. de evaluatie van MDS programma's voor symmetrische gelijkenisdata (Brouwer e.a. 1980).

Vraag 7: Het missing data probleem zal in paragraaf 6 apart aan de orde worden gesteld, dit vanwege het meer algemene karakter van dit onderwerp. In paragraaf 5 wordt dieper ingegaan op noodzakelijk uit te voeren simulaties.

3) Voor wie zijn evaluatie verslagen bestemd?

In de inleiding zijn een aantal vragen gesteld m.b.t. een databewerkings-techniek of methode. Het mag duidelijk geworden zijn dat onderzoekers pas geïnteresseerd raken in vergelijkende studies over programmatuur indien de achterliggende statistische methode bekend verondersteld mag worden.

Alleen onderzoekers met een redelijke inhoudelijke kennis m.b.t. bepaalde technieken kunnen in staat geacht worden op zinvolle wijze gebruik te maken van evaluatierapporten. Het is hierdoor ook niet nodig een theoretische beschouwing omtrent het gekozen methodologisch terrein in een evaluatierapport op te nemen. Het verwijzen naar inleidende tekstboeken is voldoende.

4) Hoe kan evaluatie worden uitgevoerd?

Deze vraag trachten we te beantwoorden aan de hand van de in paragraaf 2 opgesomde 7 problemen. In subparagrafen worden de diverse uitwerkingen gegeven.

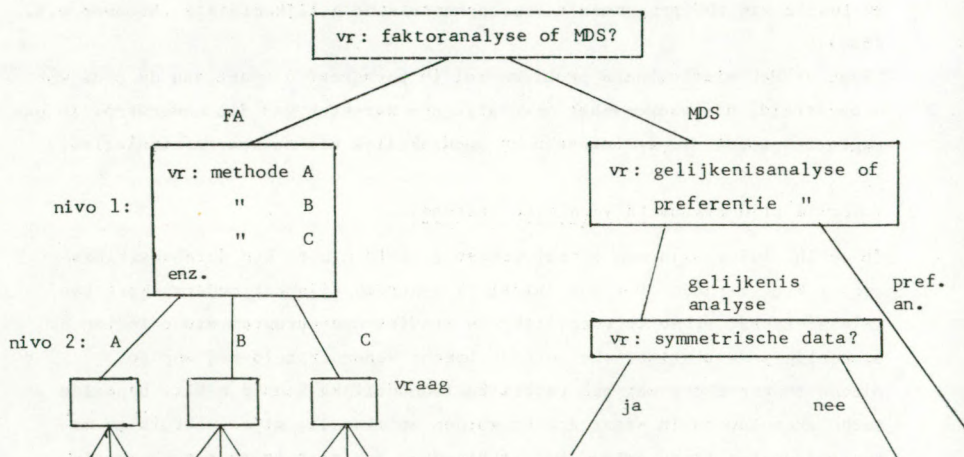
4.1 Welke programma's bestaan er voor een bepaalde methode of techniek?

De laatste jaren zijn er een aantal interactieve programma's verschenen, waarbij via een vraag en antwoord spel, uiteindelijk een of meerdere pro-

gramma's worden genoemd, die een gespecificeerd, ruw afgebakend statistisch terrein bestrijken (het TC programma TCINFO is een dergelijk voorbeeld) (Bakker, e.a. 1980).

Via het aftasten van een boomstructuur bereikt men aan het eind van elke tak een of meerdere programma's. Afhankelijk van de "diepte" waartoe men vraagt (zie fig. 1) zijn combinaties van klassen van programma's te kiezen.

figuur 1 : versimpeld voorbeeld van boomstructuur.



Voor de selectie van programma's kan men b.v. stoppen op nivo 1 of nivo 2 of eventueel op dieperliggende (sterk gespecificeerde) niveaus.

Voor de evaluatie van faktoranalytische programmatuur ligt het wellicht voor de hand alle FA programma's onder nivo 1 te selekteren. Voor analyse technieken met duidelijk sterk verschillende modellen zoals b.v. het onderscheid tussen preferentie en gelijkenisanalyse binnen MDS technieken, selekteert men eerder op een dieper nivo de relevante programma's.

In tabel I staan enkele programma's die op verschillende niveaus zijn geselekteerd genoemd.

Tabel I : voorbeeld geselecteerde programma's voor evaluatie

faktor analyse:	FACTOR	(SPSS)
	JFACTOR	(SPSS)
	P4M	(BMDP)
	FACAN	(STAP)
	EFAP	
MDS	MINISSA	
(gelijkenisanalyse	MRSCAL	
symmetrische	SMACOF	
2-wegsdata)	POLYCON	
	KYST	
	ALSCAL	

In de praktijk zal het ondoenlijk zijn veel programma's te vergelijken. Het verdient geen aanbeveling erg algemene programma's ("All purpose") in evaluatieonderzoek m.b.t. een specifieke methode op te nemen. Voor een evaluatie van faktoranalytische programmatuur is het b.v. ongewenst de programma's LISREL en G3SLS (SPSS) te onderzoeken; deze programma's zijn immers algemeen van aard (lineair structurele modellen). Alleen indien uit de praktijk blijkt dat vrij algemene programma's (zoals b.v. KYST en POLYCON) vaak voor zeer specifieke toepassingen worden gebruikt (b.v. analyse op een symmetrische gelijkenissenmatrix), dan loont het de moeite, die programma's ook in de evaluatieprocedure te betrekken.

4.2 Wat zijn de mogelijkheden van elk programma?

Wanneer eenmaal vastligt welke programma's onderzocht worden, dan moet worden nagegaan wat elk programma te bieden heeft. Aan de hand van een inventarisatie van de gebruikersmanuals worden de mogelijkheden van elk programma vergeleken. Liefst in tabel- of schemavorm worden de verschillende opties/methoden/bijzonderheden van elk programma weergegeven. In afzonderlijke overzichten staan de input- en output faciliteiten en gehanteerde methoden/technieken afgedrukt. Als voorbeeld geven we enige "mogelijkheden" tabellen uit het MDS onderzoek (Brouwer, e.a., 1980).

MDS "mogelikheden" tabellen:

Tabel II: Input options of the investigated programs

	similarity matrix	dissimilarity matrix	asymmetric data matrix	starting configuration	weights	target configuration	fixed points	LIMITS	
								number of stimuli	maximum number of dimensions
ALSCAL	x	x	x					50	6
KRST	x	x	x	x	x		x	100	6
MINISSAN	x	x		x				40	9
MINISSAX	x	x		x				70	9
NRSCAL	x	x		x				50	8
POLYCON	x	x	x	x	x	x	x	60	9
SMACOF 1		x	x	x	x			U	U

U = unlimited depending on core and computing time

Tabel IV : Output options of the investigated programs

	ALSCAL	KYST	MINISSAN	MINISSAK	MRSICAL	POLYCON	SMACOF 1
data matrix	x	x		x	x	x	x
disparities	x	x				x	
distances	x	x	x		x	x	x
weights		x				x	
residuals						x	
initial configuration			x	x	x	x	x
target configuration						x	
history		x	x	x		x	x
triangular inequalities						x	
plot initial configuration						x	x
plot target configuration						x	
plot distances - disparities	x					x	
plot data - disparities	x	x				x	
plot data - distances	x	x				x	x
m-space analysis				x			
plot stress - dimensions		x				x	

De in de tabel gehanteerde begrippen worden in het MDS rapport summier uitgelegd, immers we mogen veronderstellen dat de lezer van het rapport de theorie kent. Alleen meer specifiek aan het programma gekoppelde begrippen dragen om een uitvoeriger uitleg.

Tot nu toe is de evaluatie puur gebaseerd op schriftelijke informatie uit gebruikersmanuals eventueel aangevuld met relevante literatuur.

Het werkelijk draaien van programma's heeft nog niet plaatsgevonden. In principe moet een lezer nu al in staat zijn een voorlopige keuze uit het aangeboden programma arsenaal te doen. Alle theoretische informatie is nu bekend. Echter de praktijkervaring kan een voorlopig gedane keuze teniet doen. In theorie kan een programma prachtig werken, doch de werkelijke bevindingen, opgedaan in de praktijk, kunnen anders uitpakken. We komen hierin 4.5 op terug.

4.3 Hoe ziet de gebruikers dokumentatie eruit?

Het beoordelen van gebruikers dokumentatie heeft een vrij subjectief karakter. Wat de een ruim voldoende acht is voor de ander nauwelijks toereikend. Waar moet in het bijzonder op gelet worden bij de beoordeling van user manuals?

a) methodologische, theoretische uiteenzetting van het geprogrammeerde analysemodel. Slechts weinig manuals voldoen aan deze voorwaarde. Enkele grote programmapakketten zoals SPSS en BMDP benaderen dit ideaal, zij het niet voor elk opgenomen programma (zo bestaat de totale gebruikersdokumentatie van het SPSS programma JFACTOR uit slechts één pagina).

Voor losse applicatieprogramma's is het vaak voldoende een korte uiteenzetting van het model te geven en verder te verwijzen naar relevante literatuur. Veel gebruikte standaard programmapakketten dienen een uitvoerige theoretische inleiding te bevatten. Het gevaar bestaat echter dat niet-onderlegde gebruikers de methode leren uit de manual. Dit is natuurlijk niet de bedoeling. De inleiding dient meer als opfrisser. Een uitvoerige inleiding waarin al verwezen wordt naar keywords/opties voorkomt ellenlange omschrijvingen van de gehanteerde symbolen en notaties.

b) staan alle invoer-opties beschreven en staan ze eenduidig en begrijpelijk omschreven?

De meeste manuals geven een overzicht van alle invoermogelijkheden, er zijn althans keywords of vaste kolommen in een parameterkaart voor gereserveerd. Vaak echter zijn deze keywords onvoldoende omschreven en kunnen zij verwarrend werken (bijvoorbeeld niet adequaat gekozen namen).

Alleen door het programma een paar keer met verschillende invoer-opties te draaien, kan duidelijk worden wat de programmeur/programmabeschrijver met bepaalde keywords heeft bedoeld.

c) biedt de invoerbeschrijving niet te veel?

Het is af te raden in de gebruikers manual meer te vermelden dan het programma aan kan. Op zich is het aantrekkelijk uit te wijden over een variant van het geprogrammeerde model. Alleen zou dit niet in de gebruikershandleiding moeten gebeuren, maar in vaktijdschriften. Een handleiding dient strikt beperkt te zijn tot de geprogrammeerde methode.

Een voorbeeld uit de MDS groep:

De MINISSA handleiding (MDS(X) pakket) geeft een uitvoerige verhandeling over de tie benadering in het programma, zowel de primary als de secondary approach to ties, wordt uitgelegd. Helaas is slechts de primary approach in het programma opgenomen. Dit zelfde geldt ook m.b.t. de verschillende genoemde stress formules; alleen stress formule I is geïmplementeerd.

d) voorbeeld invoer en uitvoer.

Elke manual bevat (idealiter) een of meerdere complete invoer setups met bijbehorende uitvoer. De meest geeigende toepassingsmogelijkheden van een programma worden op deze wijze aanschouwelijk voorgesteld. De uitvoer dient bekomentariëerd te worden in die zin dat volkomen duidelijk wordt aan de gebruiker wat elk stuk deeloutput nu eigenlijk voorstelt.

Het is natuurlijk nog prettiger als in de programma output precies staat vermeldt wat alles nu betekent (voor interactieve programma's is dit nog meer noodzakelijk). In de praktijk kunnen de omschrijvingen echter onevenredig lang worden, vandaar de omschrijving in de manual. Het is voor een gebruiker uiterst aantrekkelijk dat hij, al voordat hij een programma gaat gebruiken, een idee heeft over de lay-out van tabellen en plots.

Het evalueren van de output van een programma, wat betreft de visuele aspecten, moet dus eigenlijk mogelijk zijn zonder het vooraf draaien ervan.

Met in achtneming van bovenstaande 4 evaluatiecriteria zijn binnen de MDS subgroep de eerder genoemde MDS programma's bekeken.

Een globaal overzicht wordt in tabel V gegeven.

Tabel V : output and documentation for MDS programs.

program	output plot	user documentation
ALSCAL	0	0
KYST	+	+
MINISSA	+	0
MRSCAL	+	+
POLYCON	+	+
SMACOF1	-	0

4.4 Hoe gebruikersvriendelijk is een programma?

Of een programma gebruikersvriendelijk is, is een subjectief oordeel.

Voor de ene gebruiker is dit programma het plezierigste en voor de andere dát programma. Het oordeel zal zeker afhangen van de eerdere ervaring met computerverwerking.

Enkele punten die bij het bepalen van gebruikersvriendelijkheid een rol spelen zijn:

1) de bij het programma behorende dokumentatie. Hierbij komen natuurlijk de in de vorige paragraaf genoemde criteria naar voren. Verder belangrijk is de link van het programma met de manual. De manual moet alle mogelijkheden van het programma beschrijven. Ook al kunnen we de dokumentatie afzonderlijk evalueren; bij de gebruikersvriendelijkheid van een programma speelt de dokumentatie een grote rol.

2) Invoer voor het programma.

Voor wat betreft de vereiste invoer voor een programma kan een onderscheid gemaakt worden tussen fixed format en keywords.

- Fixed format:

Via cijfers in bepaalde kolommen van de stuurkaarten wordt aangegeven welke invoer, welke methode en welke uitvoer men kiest.

Bijv. in het programma CLUSTAN geeft men met onderstaande kaart aan dat men 100 objecten heeft met 10 continue variabelen, die gestandaardiseerd moeten worden en dat de data van local file CLUSIN moeten worden gelezen.

kolom	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
	1	0	0				1	0					1						8

Een nadeel van deze methode is dat er vrij snel fouten gemaakt worden door verschuivingen in de kaart tijdens het ponsen.

- Keywords:

Keuze van opties etc. geschiedt middels een aantal woorden welke lijken op of afkortingen zijn van de gekozen opties: bijv. ROTATE=VARIMAX. Weliswaar staan veel programma's toe dat keywords door elkaar gebruikt worden, maar er is meestal toch nog steeds een bepaalde structuur vereist.

Zo kent BMDP een indeling in paragrafen, waarbinnen de keywords opgegeven dienen te worden. SPSS maakt gebruik van zgn. control cards, waarbij de keywords niet in de eerste 15 kolommen van een kaart mogen staan, omdat deze gereserveerd zijn voor de control words, die de 'job flow' bepalen.

3) Foutenafhandeling.

Heel belangrijk voor evaluatie is de vraag: "wat doet het programma als er een fout in de invoer gemaakt is?"

We kunnen hierbij een aantal fouten onderscheiden:

- Een keyword is fout of in een bepaalde kolom staat een verkeerde waarde. Het programma kan afbreken of een defaultwaarde veronderstellen en door-

gaan. De kans is echter groot dat de gebruiker de defaultwaarde niet wil en dus toch zijn run moet overdoen.

In het geval van keywords zou nog gezocht kunnen worden naar het meest gelijkende keyword, maar ook dit zal niet altijd tot het gewenste resultaat leiden.

- Keuze van strijdige opties.

Bijv. voor de behandeling van missing data, zowel listwise als pairwise deletion opgeven. Het programma dient hier vooraf op te toetsen, bij fouten afbreken en aan te geven wat er fout is.

- Foutieve data invoer.

Een voorbeeld. Als de invoer bestaat uit correlaties kan het programma toetsen of dit correlaties zijn en stoppen als er getallen zijn in absolute waarde groter dan 1.0.

Voor elke fout geldt natuurlijk dat het programma stopt met een behoorlijke foutmelding en niet zoals helaas maar al te vaak gebeurt met een van de volgende voor de gebruiker mistige foutmeldingen zoals "FTN error 65...." of "Arithmetic mode error at address ...".

De foutmeldingen dienen duidelijk te zijn zodat een gebruiker zelf tenminste nog een poging kan wagen de fout zonder hulp van derden op te lossen.

4) Weergave van de invoer.

Het is belangrijk dat de gebruiker in zijn uitvoer kan zien wat hij ingevoerd heeft en wat er gebeurd is.

Meldingen zoals die voorkomen in SPSS:

"Case ... has illegal data. Bad values were set to blank" blijken nog maar al te vaak over het hoofd gezien te worden.

5) Leesbare uitvoer.

Helaas zijn er nog steeds programma's die om welke reden dan ook zeer zuinig zijn met hun kommentaar in de uitvoer. Of een bepaald getal een multi-pele correlatie coëfficiënt is of een Spearman correlatie moet uit de uitvoer duidelijk worden.

De hierboven genoemde punten spelen waarschijnlijk niet voor iedereen een even belangrijke rol. Het is daarom moeilijk om op basis hiervan een uitspraak te doen of een programma gebruikersvriendelijk is.

Nog moeilijker is dit bij eisen als:

- het programma moet eenvoudig te gebruiken zijn
- binnen het programma moeten ook allerlei andere zaken mogelijk zijn, zoals data transformatie etc.

Wat betreft het eerste punt lijkt een beoordeling ons zeer dubieus. Het programma is eenvoudig te gebruiken als er veel defaults inzitten en de gebruiker dus nauwelijks iets meer hoeft in te voeren dan zijn data. De vraag of defaults een zegen of een ramp zijn is elders al eens beantwoord (van Nes, 1978). Verder blijkt dat een programma vaak moeilijk wordt gevonden doordat de gebruiker de terminologie behorende bij de betreffende methode niet beheerst. Dus speelt hierbij de kennis van de gebruiker een grote rol. Afgezien van computer technische zaken zal een gebruiker, die een bepaalde methode beheerst, de invoer voor een bepaald programma meestal eenvoudig zelf kunnen verzorgen. Moeilijker worden doorgaans de bijkomende zaken. Meestal moeten nog variabelen geherkodeerd worden, of zijn bepaalde selecties op het bestand nodig, enz. Een vaak gehoorde wens is dan ook dat bepaalde programma's ook data manipulatie faciliteiten dienen te krijgen. Het lijkt ons echter dat men deze kant van de programmatuur buiten de evaluatie moet houden. Wel dient men natuurlijk de invoermogelijkheden van een programma te kijken en kan men zelfs konkluderen dat het maar lastig is dat sommige MDS programma's niet zelf de afstanden berekenen of sommige faktor-analyse programma's geen correlaties. Of dit dan ook als gebruikersonvriendelijk dient te worden geklassificeerd is nog maar de vraag. Al met al is een totaal oordeel over de gebruikersvriendelijkheid moeilijk te geven en in een evaluatierapport zullen dan ook de beoordelingen op de afzonderlijke punten vermeld moeten zijn, zodat elke gebruiker zijn eigen oordeel kan vormen.

4.5 Hoe doen de programma's het in de praktijk?

Natuurlijk zal nagegaan moeten worden of het programma alles doet wat het volgens de beschrijving behoort te doen. Een groot deel van de te besteden tijd aan evaluatieonderzoek m.b.t. programmatuur, gaat in de praktische evaluatie zitten; m.a.w. het draaien van testruns.

Met de vraag naar hoe de programma's het in de praktijk doen wordt echter meer bedoeld: hoe goed doet de methode het, welk soort problemen kan de gebruiker verwachten als hij een bepaalde optie kiest, etc. Er dienen dus vergelijkingen gemaakt te worden tussen de verschillende methoden voor een bepaalde techniek en de verschillende alternatieven binnen een methode. De gebruiker moet kiezen en het raadplegen van een evaluatierapport zal hopelijk deze keuze vergemakkelijken.

Een aantal voorbeelden van keuzeproblemen:

* FAKTORANALYSE

Op de eerste plaats zal de gebruiker moeten bepalen welke methode hij wil: principale komponentenanalyse, principale faktoranalyse met of zonder iteraties, alpha faktoranalyse, Jöreskog analyse etc. Daarna zal hij het aantal factoren moeten bepalen, welke rotatie en op welke manier de faktorscore coëfficiënten berekend moeten worden.

* CLUSTERANALYSE

Welke clusteranalyse, welke associatiemaat, wel of niet standaardisatie, aantal clusters.

* MDS

Welke startconfiguratie, welke metriek, welke tie benadering, welke missing data behandeling, welke stressformule, het aantal dimensies.

In het evaluatierapport moeten aanbevelingen staan over welke optie men het beste kan kiezen bij welk soort data. Bij een veelvuldig gebruikte techniek als faktoranalyse kan men een groot aantal aanbevelingen in de literatuur vinden. Dit geldt in mindere mate voor CLUSTER analyse en in nog mindere mate voor MDS. Het verdient in elk geval aanbeveling om de bevindingen uit de literatuur op te nemen. Men zal zich in sommige gevallen wenden tot simulaties, waarbij men weet wat de oplossing moet zijn en men dus kan beoordelen in hoeverre de oplossing van het programma hieraan voldoet. (Op de opzet van simulaties komen we in de volgende paragraaf terug.)

We zullen weer aan de hand van het MDS evaluatierapport enkele voorbeelden hiervan geven:

- Startconfiguraties

Na een opsomming van de mogelijke startconfiguraties en een summier uitleg volgen de resultaten van een simulatie met 20 random configuraties van 10 punten in twee dimensies (de verwachte stress hierbij is dus nul).

Tabel VI : mean stress value, minimum and maximum stress value of 20 random configurations in two dimensions using 5 different random configurations and the torsca and minissa starting configurations.

	mean	minimum	maximum
random=1	.120	.005	.633
random=2	.042	.003	.668
random=3	.057	.001	.530
random=4	.051	.003	.321
random=5	.111	.004	.743
torsca	.001	.000	.005
minissa	.007	.002	.009

De conclusie hieruit is dat TORSKA duidelijk de beste oplossing te zien geeft en dat runs met random startconfiguraties vaak resulteren in lokale minima.

- Minkowski metriek

In het rapport wordt eerst de formule gegeven van de minkowski metrie-ken, waarna men zich verder beperkt tot de 3 meest bekende metrie-ken:

- * de city block metriek
- * de euclidische metriek
- * de dominantie metriek

Over deze metrie-ken worden enkele bevindingen uit de literatuur aangehaald, waarna ook hier de resultaten van enkele simulatieruns met random configuraties volgen.

Tabel VII : Mean stress values for twenty random configurations when the distance is computed using one of the minkowsky metrics and analysis is done with the same metric or one of the other metrics.

VIIa: Used program: KYST nonmetric.

distance computed using the	analysis done with		
	city block	Euclidean	dominance
city block metric	.075	.057	.021
euclidean metric	.053	.001	.051
dominance metric	.050	.070	.061

VIIb : Used program KYST metric.

distance computed using the	analysis done with		
	city block	Euclidean	dominance
city block metric	.081	.129	.047
euclidean metric	.119	.007	.114
dominance metric	.120	.134	.098

VIIc : Used program MINISSAN

distance computed using the	analysis done with		
	city block	euclidean	dominance
city block metric	.046	.032	.013
euclidean metric	.026	.000	.021
dominance metric	.048	.032	.025

De conclusie m.b.t. de te kiezen metriek luidt dat als de onderzoeker de onderliggende metriek niet kent hij het beste de euclidische metriek kan kiezen, omdat deze minder kans geeft op lokale minima en gedegene-

reerde oplossingen en ook omdat, al is de onderliggende metriek in werkelijkheid niet de euclidische, deze toch de echte, aan de data ten grondslag liggende, configuratie zeer goed benadert.

Noot:

Bij de MDS programma-evaluatie is niet speciaal gelet op de verwerkings-snelheid. Reden: de verschillende programma's ontliepen elkaar niet veel m.b.t. deze tijdsfaktor (nb: dit geldt voor de relatief kleine datasets die in het onderzoek zijn geanalyseerd). Het is echter aan te bevelen in algemene programma-evaluatie wel degelijk met de verbruikte computertijd rekening te houden.

4.6 Worden model assumpties vooraf getoetst?

Deze vraag moet bij de bestaande programmatuur vaak met neen beantwoord worden. De vraag hiernaar wordt echter steeds meer gesteld, zodat programma's die model assumpties toetsen extra pluspunten dienen te verkrijgen. Het verdient aanbeveling ontwikkelaars van programmatuur te wijzen op de noodzaak model-voorwaarden te checken. Vaak is dit echter praktisch onuitvoerbaar. Vooral in multivariate modellen zijn modeltoetsen dikwijls uiterst ingewikkeld. Eenvoudige assumpties zoals bv. lineariteit, meetniveau en normaliteit zijn echter relatief eenvoudig in veel statistische programma's in te bouwen.

Het programma dient af te breken, of althans een waarschuwing te geven, indien de assumpties geweld wordt aangedaan. Overigens dienen ook robuustheidsstudies m.b.t. modelassumpties bij programma-evaluatie aan de orde te komen.

5) Simulaties.

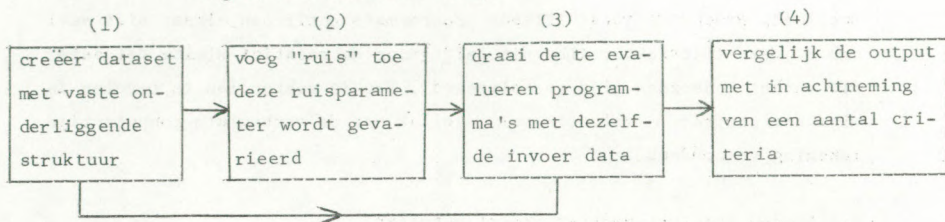
Voor programma-evaluatie is het onontbeerlijk gebruik te maken van statistische simulatietechnieken. De korrekte en efficiënte werking van een computerprogramma is te onderzoeken via het invoeren van bekende datasets. Hieronder verstaan we datasets met een vooraf bekende structuur.

De onderzoeker weet al van te voren wat het programma idealiter aan resultaten zou moeten geven. Er wordt bijv. een correlatiematrix van 10 variabelen ingevoerd, waarin al een 2-faktorstructuur is aangebracht. Het zal duidelijk zijn dat een faktoranalyse programma een dergelijke structuur terug zal moeten vinden. Bij programma-evaluatie is gebleken dat dit lang

niet altijd het geval is. Er zijn situaties bekend waarin duidelijk afwijkende resultaten zijn gevonden.

Hoe ziet nu een algemeen simulatiedesign m.b.t. programmatuurevaluatie er uit? In figuur 2 is een schematische weergave gegeven.

figuur 2 : simulatie design



Hoe zijn nu in concreto de datasets uit (1) te bepalen?

Afhankelijk van de methode of het model dat in de te evalueren programma's is geïmplementeerd, wordt een keuze gedaan. Enkele voorbeelden:

a) voor faktor analytische modellen:

- ga uit van b.v. p manifeste variabelen en r onderliggende latente factoren.
- ken ladingen toe m.a.w. geef een ladingenmatrix aan.
- korreleer eventueel de factoren.
- bereken op grond van het bovenstaande de variantie/covariantiematrix (in de praktijk kan dit b.v. eenvoudig met hulpprogramma's gebeuren (vermenigvuldigen van matrices) of m.b.v. LISREL) en zet deze matrix om tot een korrelatiematrix.

NB: de berekende matrix is natuurlijk singulier (meer variabelen dan factoren).

De eerste simulatieruns zijn nu uit te voeren met als kernvraag: Hoe reageren FA programma's op singuliere matrix invoer?

- trek ruwe data uit een multivariate normale verdeling (of een andere verdeling indien nadruk op de robuustheid van de methode ligt) met de gegeven correlatiematrix als invoer. Dus trek uit $N(0, \Sigma)$ waarin 0 een nulvektor (lengte p) en Σ de berekende correlatiematrix.

Deze trekkingprocedure is vrij eenvoudig m.b.v. IMSL routines uit te voeren.

NB: soms is het noodzakelijk Σ eerst te "smoothen" d.w.z. de matrix positief definit te maken.

b) regressiemodellen:

- stel een regressie vergelijking op b.v.,

$$Y = a + b_1 X_1 + b_2 X_2 + e \quad (1)$$

met vaste bepaalde a, b_1 en b_2 waarden.

- trek X_1, X_2 en e uit (normale) verdelingen.
- korreleer X_1 en X_2 via bv. een transformatie op X_2

$$\text{bv. } X'_2 = X_2 \sqrt{1 - \rho^2} + \rho X_1 \quad (2)$$

waarbij ρ de populatiekorrelatie is tussen X'_2 en X_1 , en vervang X_2 door X'_2 .

- bereken de Y scores, de ruwe dataset is nu compleet dwz Y_i , X_{1i} en X_{2i} scores voor $i=1, N$ personen.

c) MDS modellen:

- de eerste stap voor het specificeren van een datastructuur voor MDS modellen (zoals de MDS subgroep heeft onderzocht) is uiterst eenvoudig; teken een puntenconfiguratie in 1, 2 of 3 dimensies (voor meer dimensies worden alleen de coördinaatwaarden gegeven) en bereken de afstanden tussen de punten (dit via de minkovsky afstandenmetriek). Het resultaat is een symmetrische afstandenmatrix.

We willen nu onze aandacht richten op fase (2).

Hier wordt al dan niet "ruis" toegevoegd aan de "perfekte" datasets.

Voor de 3 genoemde voorbeelden schetsen we hoe dit kan plaatsvinden.

a) FA modellen:

- via LISREL zijn error termen in het model op te nemen en is aldus een covariantie matrix inclusief error te berekenen.

b) regressiemodellen:

- het variëren van de error component in (1) is een eenvoudige zaak.
- multicollineariteit is te introduceren door ρ in (2) hoog genoeg te kiezen.
- ook is het mogelijk met de prediktoren te correleren. Op deze wijze is de invloed van inbreuken op model-assumpties te onderzoeken (robustheidsonderzoek dus).

c) MDS modellen:

- na constructie van de perfekte symmetrische afstandenmatrix, wordt meestal ruis in de data opgenomen door aan enkele (of zelfs alle) afstanden in de matrix een errorcomponent toe te voegen.

Stel we hebben een afstanden matrix $D=(d_{ij})$; de ruis wordt nu aldus geïntroduceerd:

$$d'_{ij} = d_{ij} + e_{ij} \quad \text{met } e_{ij} \approx N(0, \sigma^2)$$

voor bepaalde i en j waarden.

Via variatie van σ^2 is de ruisparameter systematisch te controleren.

De nieuwe "afstanden" matrix D' bevat geen werkelijke afstanden meer. (De driehoeksongelijkheid hoeft bv. niet meer op te gaan.) Een perfecte afbeelding van D' in de vooraf gespecificeerde r -dim ruimte is ook niet meer mogelijk. Spence en Graef (1974) hebben o.a. via deze methode hun MSPACE routine ontwikkeld.

Opmerking: Bij de evaluatie van de eerder genoemde MDS programma's is bovenstaande bewerking niet uitgevoerd. Slechts perfecte datasets zijn geanalyseerd, omdat we primair geïnteresseerd waren in opties, zoals tie-behandeling, en start-configuraties. Hiervoor is het wenselijk met perfecte datasets te werken. Dit neemt echter niet weg dat bij een volledig evaluatieonderzoek wel gewerkt dient te worden met ruis in de data.

Dat sluit meer aan bij de werkelijkheid. Reële datasets bevatten doorgaans een bepaald percentage "ruis". Alleen is vrijwel altijd onduidelijk hoe hoog dit percentage is.

In fase (4) worden vergelijkingen gemaakt op grond van criteria. Wat zijn die criteria?

Ook nu weer lijkt het zinvol aan de hand van de 3 genoemde voorbeelden deze criteria nader te belichten.

a) FA modellen:

- Doordat de structuur in de perfecte datasets vastligt en daarmee ook de ladingen en interkorrelaties tussen de factoren (nb: bij niet-perfekte datasets ligt ook de error-variantie vast), zijn de geschatte parameters direkt te vergelijken met de invoerparameters.

Voor elk programma zijn bv. verschillende mean squared errors (MSE) te berekenen.

De fit van het "ruis" model t.o.v. de fit in het perfecte model is eveneens te vergelijken. In tabelvorm is voor elk onderzocht programma de MSE voor de geschatte parameters, de fit, de residuen matrix, enz. te vermelden.

Bijzondere bevindingen dienen in het evaluatie verslag te worden opgenomen bv. het feit dat niet alle programma's eenzelfde aantal factoren trekken.

b) regressie modellen:

- In deze situatie zijn schattingen van a , b_1 en b_2 te vergelijken met de ingevoerde waarden hiervoor. Verder zijn berekende multiële correlaties onderling te vergelijken, of de gemiddelde afwijking tussen de geschatte en de werkelijke Y waarden (de residuen dus).

c) MDS modellen:

- De vergelijkingskriteria voor MDS evaluatie zijn minder duidelijk vast te stellen. Dit vindt zijn oorzaak in het feit dat in tegenstelling tot FA en regressie modellen in MDS modellen niet direkt parameterwaarden worden geschat.

Natuurlijk zijn de gevonden schaalwaarden (de eindconfiguratie) dwz coördinaatwaarden voor elk stimuluspunt op te vatten als parameters, maar ze hebben een minder eenduidige betekenis: veel schaaltransformaties zijn mogelijk zodat onderlinge vergelijking uiterst lastig wordt. We komen eenvoudiger tot evaluatiecriteria, wanneer we het primaire doel van MDS analyse voor ogen houden: vind een afbeelding van de invoer "afstanden" matrix in een ruimte met een zo klein mogelijk aantal dimensies en probeer zonodig die dimensies te interpreteren. Het gaat er hierbij om of de uitvoerconfiguratie er "ongeveer" net zo uit ziet als de invoerconfiguratie.

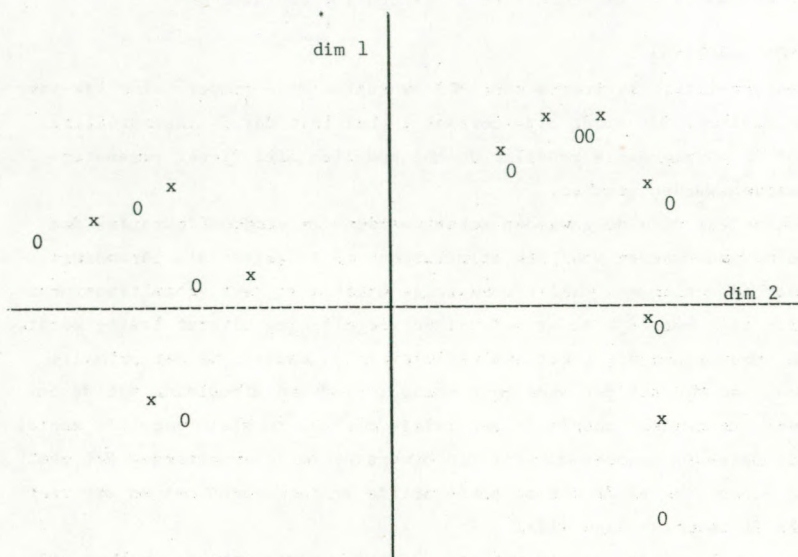
De vergelijkingskriteria zullen een subjectiever karakter hebben. Dit neemt echter niet weg dat er hardere criteria zijn om configuraties te vergelijken (zie bv. de laatste paragraaf van het MDS rapport).

In onze MDS evaluatie hebben we primair gebruik gemaakt van vergelijking op grond van gevonden stressmaten (indikatie voor modelfit). Daarnaast is op vrij intuïtieve wijze gekeken naar de overeenstemming tussen in- en uitvoerpuntenconfiguratie. In verbale omschrijvingen zijn die laatste vergelijkingen weergegeven. Een voorbeeld ter illustratie: "Alle onderzochte MDS programma's vonden de clusterstructuur in de invoermatrix terug, zij het dat voor programma XX binnen de clusters de stimuli dichter op een gehoopt lagen".

Naast visuele inspectie van twee of meer eindconfiguraties leek het nuttig (in die situatie waarin dataplots op dezelfde wijze genormeerd zijn) verschillende plots op elkaar af te drukken: afwijkende stimuli zijn zo simpel op te sporen.

In figuur 3 is een fiktief voorbeeld te zien.

figuur 3 : eindconfiguratie voor 2 MDS programma's met
dezelfde data invoer.



0 = stimulus weergave via programma A

x = " " " " B

Wellicht ten overvloede wijzen we erop dat naast de gesimuleerde datasets ook reële, empirische datasets in de evaluatieprocedure moeten worden opgenomen. Vooral heranalyse van uit de literatuur bekende datasets (die al vaak op verschillende wijzen zijn geanalyseerd) kunnen uiterst nuttige informatie verschaffen.

Een probleem is echter om bekende empirische datasets te vinden, waarvoor mag worden aangenomen dat ze geschikt zijn voor analyse middels de onderzochte modellen. De eis dat de structuur in de data bekend moet zijn is een uiterst zware.

6) Missing data problemen.

Tot voor kort werd bij de ontwikkeling van computerprogramma's niet of nauwelijks rekening gehouden met het feit dat onderzoeksmateriaal lang niet altijd volledig is. In vrijwel elke reële dataset komen ontbrekende scores voor.

Voor een adequate programma-evaluatie is het noodzakelijk aan dit probleem aandacht te schenken. Tegenwoordig bevatten recentere programma's (pakketten) vaak opties om met missing data om te gaan. De vraag is nu hoe sterk deze opties verschillen, en wat men moet doen indien een programma geen opties kent voor de aanpak van missing data.

Besselink (1980) constateerde bv. dat een FA op een inkomplete dataset regelmatig een totaal ander FA model fit dan dezelfde analyse op de complete invoer dataset.

Afhankelijk van de missing data optie (en natuurlijk het missing data patroon) worden FA modellen gefit met sterk verschillende parameterwaarden.

Het evaluatiedesign voor de missing data behandeling is geheel analoog aan dat gegeven in figuur 2. In fase (2) wordt nu geen ruis geïntroduceerd maar bepaalde missing data patronen (al dan niet random).

Door weer het "volledige" model te vergelijken met het "onvolledige" is het mogelijk uitspraken te doen over de effecten van verschillende missing data opties en de vraag hoe goed of slecht bepaalde methoden omgaan met ontbrekende.

Een uitgewerkt voorbeeld uit het MDS onderzoek.

Sommige onderzochte MDS programma's hebben opties voor de behandeling van missing data, nl:

- a) ontbrekende cellen in de invoer matrix doen niet mee in het iteratieproces (voor minimalisering van de stresswaarden).
- b) voor ontbrekende cellen wordt het totaal gemiddelde of het rij of kolom-gemiddelde ingevuld.

In ons evaluatieonderzoek hebben we drie benaderingen gekozen, te weten:

- M1) geen schattingen van ontbrekende elementen.

(D.w.z. alle waarden, dus ook bv. nullen, worden als niet ontbrekend opgevat). Een aantal MDS programma's werkt uitsluitend op deze wijze.

- M2) weglaten van ontbrekende gegevens uit de analyse.

De berekende stress is uitsluitend gebaseerd op de bekende data elementen.

- M3) invullen van het totaal gemiddelde voor de ontbrekende elementen.

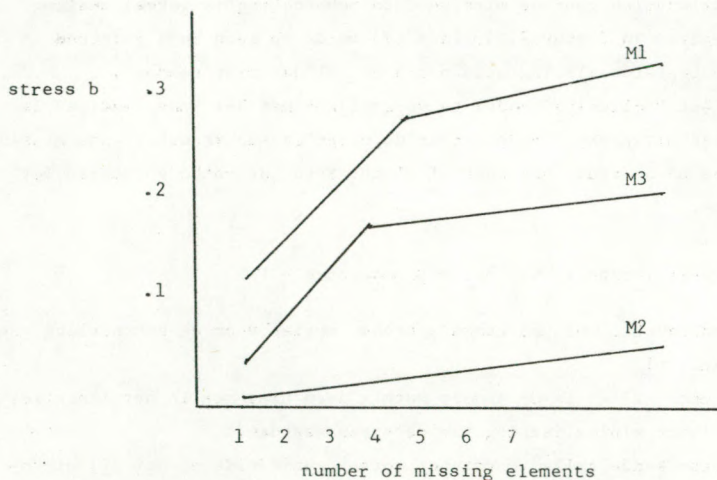
Een vast percentage random missing data is bij elke simulatie run geïntroduceerd. Gekozen is voor een vaste configuratie van 10 punten en twee dimensies.

De runs zijn uitgevoerd m.b.v. het programma KYST waarbij optimale opties zijn gekozen. (Nb: uit eerder simulatieonderzoek zijn de optimale opties bekend geworden.)

Als vergelijkingskriterium is gekozen voor de stress (sform2) berekend tussen de originele afstanden (dus zonder missing data) en de gevonden, gefitte afstanden (met missing data).

In figuur 4 staan de bevindingen.

figuur 4 : interactions effect between method and
n of missing data.



Het niet mee laten tellen van ontbrekende data elementen levert de beste resultaten op.

Dit is intuïtief, voor de onderzochte situatie, ook wel duidelijk; immers de 45 ($\frac{1}{2} \times 10(10-1)$) afstanden tussen de 10 punten leveren de restricties voor de afbeelding van de punten in de 2-dim ruimte.

Het kan geen kwaad 15 tot 20% random elementen weg te laten; er zijn dan immers minder restricties; met als gevolg een betere fit in de 2-dim ruimte (de oorspronkelijke invoer-configuratie wordt echter bij veel ontbrekende data elementen in de invoer data matrix niet meer terug gevonden: zie ook de M2 lijn in de figuur).

7) Aanbeveling

Evaluatie van statistische programmatuur zou op de volgende wijze kunnen plaatsvinden:

- 1) stel "mogelijkheden" tabellen op. D.w.z. overzichtstabellen met invoer- en uitvoer opties en methoden van elk onderzocht programma.
- 2) kies te analyseren datasets.
 - a) uit literatuur bekende datasets met bekende eigenschappen/structuur.
 - b) genereer kunstmatige datasets met bekende onderliggende structuur.
 - c) gebruik extreme datasets (ook weer via simulatie technieken te genereren).
- 3) analyseer de datasets; kies voor elk programma dezelfde data invoer.
 - a) gebruik veel opties en combinaties van opties.
 - b) vergelijk output (aan de hand van vooraf bepaalde criteria; voor elke (multivariate) techniek doorgaans andere criteria).
 - c) let op rekennauwkeurigheid.
 - d) introduceer missing data in bestand en bekijk de effecten/verstoringen die hierdoor ontstaan.
- 4) geef de resultaten uit 3 indien mogelijk in tabel vorm weer (vergelijkbaar met de bekende tabellen uit de consumentengids).
- 5) merkwaardige bevindingen uit 3 kunnen eventueel nader uitgewerkt worden. Probeer achter de oorzaken hiervan te komen, dit laatste is speciaal van nut voor programma/algorithmen verbetering.

De praktijk heeft tot nu toe uitgewezen dat een evaluatie onderzoek (zoals hierboven omschreven staat) binnen het kader van de CSP subgroepen moeilijk uitvoerbaar is. De vrijblijvendheid van de individuele subgroepsleden is te groot om een ambitieus plan te kunnen uitwerken. Men kan er zelf niet teveel tijd in steken. Wellicht is het mogelijk via bv. de NOSMO* computer-programmatuur-groep van de sectie "methoden van data analyse" subsidie aan te vragen voor programmatuur-evaluatie onderzoek. Vast staat in elk geval dat gebruikers van statistische programmatuur erg geholpen zouden zijn met beschikbare evaluatierapporten. Voor consultatie m.b.t. programmatuur is programma-evaluatie onontbeerlijk.

* Nederlandse Organisatie voor Sociaal Methodologisch Onderzoek

Literatuur

Bakker, D., Brouwer, U. en Debets, P. "TCINFO: een automatische programma-detector". MDN 5.1 febr. 1980.

Besselink, J. "Het terugvinden van het oorspronkelijke model door verschillende schattingsmethodes voor de korrelatie-coëfficiënt. In: "Random Missing Data". Stageverslag methodenleer, juni 1980.

Brouwer, U., Debets, P. en van Schuur, W. (1980).

Multidimensional Scaling: some evaluations concerning the distance model in MDS programs, using symmetric input matrices.
Technisch Centrum, publikatie nr. E53.

Francis, I (1979). A comparative Review of Statistical Software. IASC, Voorburg, Netherlands.

van Meurs, A. (ed.), (1978). Cluster-Analyse.
Geografisch Instituut, R.U. Utrecht.

van Nes, F. (ed.), (1978). "Defaults, een zegen of een ramp" in: MDN, jaargang 3, nr.3, pag. 63-75.

Spence en Graef (1974). The determination of the underlying dimensionality of an empirically obtained matrix of proximities. Mult.Beh.Res., 9, pag. 331-341.