

Toepassing van UniNorms bij de analyse van cursusevaluaties.

Bachelorscriptie Informatica & Economie

Maarten Tijhof

275772mt

8 juni 2006

Onder begeleiding van dr. R. Potharst

1 – Probleemstelling.....	4
2 – Economische achtergrond	5
3 – Inleiding UniNorms	6
4 – Theorie UniNorms.....	9
5 – Beschrijving onderzoek	19
6 - Beschrijving data.....	26
6.1 - Oorsprong data.....	26
6.2 - Kwaliteit en kwantiteit.....	26
6.2.1 - Dataset A	27
6.2.2 - Dataset B	27
6.2.3 - Dataset C.....	27
7 – Toepassing op casussen.....	28
7.1 - Dataset A	28
7.2 – Dataset B.....	30
7.3 – Dataset C.....	31
8 – Vergelijking met andere methoden.....	33
9 – Conclusie.....	35
10 – Literatuurlijst	37
11 – Bijlagen	39
11.1 – Implementatie	39
11.2 – Programmacode.....	42
11.2.1 – UniNorms.UniNorm.....	42
11.2.2 – UniNorms.Process.....	43
11.2.3 – UniNorms.PreProcessor.....	46
11.2.4 – UniNorms.Response.....	49
11.2.5 – UniNorms.Calculator	52
11.2.6 – Utils.RandomCreator.....	53
11.3 – Datasets.....	55
11.3.1 – Dataset A	55
11.3.2 – Dataset B	55
11.3.3 – Dataset C	56

1 – Probleemstelling

Als manager is het zeer belangrijk om alle acties zo efficiënt mogelijk uit te voeren. Dit bespaart geld, tijd en zorgt er in de regel voor dat de netto winst groter wordt.

Zo ook bij klanttevredenheid, de verantwoordelijke persoon wil een zo groot mogelijke klanttevredenheid en het liefst wil hij of zij dat tegen zo laag mogelijke kosten, zo niet helemaal zonder kosten. Hierdoor is het belangrijk om juiste informatie te verkrijgen over wat de klanten op dit moment vinden van het product of de dienst die aangeboden wordt en dat er valide en effectieve conclusies getrokken worden uit de gegevens die voorhanden zijn.

Klanttevredenheid wordt over het algemeen uitgedrukt in een rapportcijfer zodat snel inzichtelijk gemaakt kan worden wat de gemiddelde klant denkt over het product als geheel. Klanttevredenheid wordt in de regel gemeten met een (schriftelijke) enquête waarbij niet alleen wordt gevraagd naar de mening over het product of de dienst in het geheel, maar er worden ook vragen gesteld aan de geselecteerde klanten over bepaalde onderdelen van het product of de dienst. Ook hier dienen beoordelingen aan gegeven te worden, waarbij er ook vaak wordt gemeten op een numerieke schaal. Deze numerieke beoordelingen lenen zich bij uitstek voor aggregatie en interpretatie.

Er is al een behoorlijk aantal methoden ontwikkeld om op basis van productbeoordelingen de zwakke en sterke punten van een product of dienst te identificeren maar vaak hebben deze technieken (ernstige) tekortkomingen. Vanhoof et al. (2005) (zie [15]) hebben een alternatief gepresenteerd in de vorm van een universele aggregatieoperator, ook wel bekend onder de term “uninorm”. Dit onderzoek is een vervolgonderzoek naar de praktische toepasbaarheid van deze techniek, waarbij er een dataset wordt gebruikt zoals deze ook in de praktijk gevonden kan worden. Vergelijkbare datasets zijn de gegevens die voortkomen uit onderzoek onder klanten naar aanleiding van een aanschaf, of bijvoorbeeld resultaten van vragenlijsten die klanten voorgelegd worden wanneer men overstapt van de ene aanbieder naar de andere.

2 – Economische achtergrond

Omzetten staan of vallen met de klanttevredenheid van de consumenten, zeker ook in het huidige tijdperk wanneer er eigenlijk altijd wel een concurrent actief is, al dan niet via de moderne communicatietechnieken als het internet. Vandaar dat het meten van klanttevredenheid en het actief op problemen en hiaten inspelen tegenwoordig een must is voor bedrijven in alle takken.

Met deze metingen kwamen de analysemethoden en de queeste naar de meest effectieve methode om de gegevens te analyseren om een voorsprong te verkrijgen op eventuele concurrenten. In onder andere hoofdstuk 3 zullen een aantal van deze analysetechnieken aangestipt worden, evenals een techniek die centraal staat in dit onderzoek: analyse met behulp van uninorms.

In het algemeen kan een analyse van klanttevredenheidsgegevens goed weergeven over welk aspect de klanten niet zo goed te spreken zijn, en wat er juist goed uit de verf komt. Nu is dit slechts één deel van de gegevens die idealiter bekend zouden zijn, het andere deel is hoeveel invloed een bepaald aspect en de beoordeling van de klant op het eindoordeel heeft. Voor zover bekend is er geen methode die deze informatie afdoende bekend kan maken en is er gezocht naar een methode die het wel voor elkaar kon krijgen. Analyse met behulp van uninorms is, getuige het betoog later in dit onderzoek de methode die hier bij uitstek geschikt voor is.

Wanneer zowel de beoordeling als de invloed van de betreffende beoordeling bekend is, is het relatief eenvoudig om deze gegevens te gebruiken om de dienstverlening te verbeteren. De kunst is het, om zo kostenefficiënt mogelijk, een zo groot mogelijke verbetering van de gemiddelde waardering te verkrijgen.

De klassieke aanpak van bedrijven was om de klanttevredenheid simpelweg te maximaliseren, wat vaak tot onnodige uitgaven zorgde. Zo werd er vaak veel geïnvesteerd in aspecten die later van weinig tot geen invloed bleken op het eindoordeel, wat in essentie een niet-ideale besteding van kostbare bronnen was. Om dergelijke verspilling te voorkomen kan men veel beter op zoek gaan naar een methode om de bronnen zo effectief mogelijk, en dus geoptimaliseerd, te besteden om uiteindelijk een zo maximaal mogelijke gemiddelde stijging van de klanttevredenheid te verkrijgen.

Het gaat er dus niet om dat de tevredenheid op alle aspecten stijgt, zolang de tevredenheid op de meest bepalende aspecten maar toeneemt. In een later stadium kan er nog wel tot aspecttevredenheidsmaximalisatie overgegaan worden, maar het heeft zeker geen prioriteit.

3 – Inleiding UniNorms

Wanneer klanten van een product of dienst een oordeel moeten vellen doen zij dit op basis van hun ervaringen met het product of dienst, en baseren zij deze eindwaardering op waarderingen voor verscheidene aspecten van het product of dienst. Deze eindwaardering is geen simpel gewogen gemiddelde, maar is per persoon en per product of dienst verschillend.

Deze inzichten zijn een geaccepteerd gemeengoed geworden in de marketingwereld, zie [11] en [13]. Een voorbeeld kan zijn de klanttevredenheid over een verzekering.

Een klant van een van de grote zorgverzekeraars in Nederland kan erg tevreden zijn over de dekking die de polis biedt en over de maandelijkse premies van de verzekering, maar kan bij een vraag de verzekeraar slecht bereiken en wordt dan ook nog eens geconfronteerd met een onvriendelijke medewerker en een hoge telefoonrekening door een betaald servicenummer van de verzekeraar in kwestie.

De klant in kwestie wil graag een verzekeraar die goed bereikbaar is, en zal nu zijn eigen verzekeraar een slechte beoordeling geven omdat deze op een voor de klant belangrijke aspect als bereikbaarheid niet voldoende scoort, ondanks dat zowel de premie als de dekking zeer naar tevredenheid zijn.

Het is zeer wel mogelijk dat een andere klant bij dezelfde verzekeraar en vergelijkbare ervaringen een positiever of zelfs een ronduit positief oordeel velt. Dit is mogelijk doordat deze tweede klant kennelijk (veel) meer waarde hecht aan de dekking en/of die prijs/kwaliteit verhouding en minder hangt aan bereikbaarheid.

Een zelfde situatie doet zich voor als deze laatste klant bijvoorbeeld ook een financieel product afneemt bij dezelfde verzekeraar en identieke ervaringen met dat product heeft opgedaan. (Goede prijs/kwaliteit verhouding, slechte service en slechte bereikbaarheid) Het is niet ondenkbaar dat de klant nu zijn verzekeraar een slechte score geeft voor het financiële product, simpelweg omdat de klant bij het financiële product veel waarde hecht aan goede service zodat hij met een vraag altijd terecht kan.

Eindwaarderingen zijn dus zowel per klant als per product verschillend door persoonlijke voorkeuren van de klant zelf.

Wanneer een klant zijn eindoordeel velt over een product of dienst doet hij dit dus aan de hand van tussenwaarderingen van verscheidene aspecten. Tijdens dit ‘samenvoegen’ van de deelscores kunnen er verschillende effecten optreden die het eindoordeel zowel positief als negatief kunnen beïnvloeden. De twee belangrijkste aspecten zijn ‘reinforcement’ en ‘anchoring and adjustment’.

Wanneer een klant een product op nagenoeg ieder punt slecht zou beoordelen ligt het redelijk voor de hand dat ook het eindoordeel slecht is.

Wat opvalt, is dat de beoordeling slechter is dan men zou verwachten. Het eindoordeel ligt lager dan een gewogen gemiddelde wat in de regel toegepast zou worden door de klant.

Dit heeft te maken met het feit dat de slechte deelbeoordelingen elkaar versterken, en zo dus cumulatief mee gaan wegen in het eindoordeel. De klant is dan extra negatief over het product of de dienst omdat deze op meerdere punten simpelweg tekortschiet. Wat ook duidelijk zichtbaar is, is dat wanneer een aspect van het product tekortschiet de klant in toenemende mate negatief is in de deelbeoordeling.

De waardering neemt meer dan lineair af wanneer de prestatie lineair af zou nemen, de negatieve deelbeoordelingen versterken elkaar. Dit effect van ‘reinforcement’ wordt waarschijnlijk veroorzaakt door het contrast wat de klant meent te zien met een gemiddelde prestatie van het aspect. ‘Reinforcement’ is voor het eerst door Yager en Rybalov in 1998 (zie [18]) volledig besproken. De genoemde effecten treden ook op bij een toenemende prestatie en bijbehorende positieve beoordeling van het product door de klant.

Als een klant een overall beoordeling vormt over een product zal hij dit zoals gezegd doen op basis van de prestaties van het product op verschillende onderdelen. Zo zal de klant van verscheidene attributen van een product een beoordeling maken, en deze, al dan niet bewust, per attribuut vergelijken met de score die de klant vindt dat het attribuut moet hebben.

De klant heeft dus per attribuut een idee van hoe goed deze hoort te scoren, en baseert de daadwerkelijke beoordeling van het attribuut op dit van tevoren gemaakte oordeel. Dit oordeel heet ook wel een ‘anchor’, en is dus de waarde waar de klant van uit gaat. Op basis van deze waarde bepaalt de klant of het attribuut goed (beter dan de ‘anchor’) of slecht (slechter dan de ‘anchor’) presteert.

Deze ‘anchor’-waarde kan gebaseerd zijn op voorgaande ervaringen met het product of voorgaande ervaringen met een vergelijkbaar product, of zelfs gebaseerd zijn op ervaringen van anderen met het product of een vergelijkbaar product.

Net als voor de attributen van een product heeft de klant ook een ‘anchor’-waarde voor het product als geheel, welke voor iedere klant en voor ieder product verschillend is of kan zijn.

Dit alles kan het best duidelijk gemaakt worden aan de hand van een voorbeeld.

Klant	s_1	s_2	s_3	S	Effect
A	5	3	6	4	Compensatie
B	5	6	5	7	“Upward reinforcement”
C	3	3	4	2	“Downward reinforcement”
D1	5	7	6	6	Individu i
D2	5	7	6	5	Individu j

Tabel 3.1

Hierboven is voor een vijftal klanten de evaluatie van product X op een drietal punten weergegeven. De kolommen s_1 tot en met s_3 geven waarderingen weer van aspecten van product X terwijl kolom S de eindwaardering van die klant weergeeft.

Het voorbeeld van klant A is de meest eenvoudige van de geschetste voorbeelden, hier is de eindwaardering een simpel gewogen gemiddelde van de deelwaarderingen. De hogere en lagere deelwaarderingen compenseren elkaar waardoor een licht vertekend beeld zou kunnen ontstaan.

Klant B heeft een interessantere evaluatie gegeven, hier is goed zichtbaar dat de deelwaarderingen elkaar in een positieve zin versterken. Het eindoordeel is hoger dan de hoogste deelwaardering wat duidt op een heel lage verwachtingswaarde van de klant, immers deze is erg blij met de prestatie van het product op alle aspecten, wat leidt tot een hoge eindwaardering. Dit fenomeen is in de literatuur beter bekend als ‘upward reinforcement’.

Voor klant C geldt een zelfde redenering, zij het andersom. De eindwaardering is lager dan de laagste deelwaardering wat duidt op een, in de ogen van de klant, slechte prestatie van product X op alle punten. Klanten D_1 en D_2 hebben een lastigere interpretatie, hier geven beiden identieke tussenbeoordelingen maar met een verschillende eindbeoordeling. Dit kan duiden op verschillende verwachtingen ten aanzien van het product.

Dit alles vraagt om een manier om op een dusdanige manier deelwaarderingen op te tellen tot een eindoordeel conform de wijze van een willekeurige consument. Eerder is besproken dat een eenvoudig gewogen gemiddelde in de meeste gevallen geen correcte uitkomst geeft, zodat er gezocht moet worden naar een methode die deze ‘optellingen’ wél voor iedere klant een accuraat geeft.

Er zijn in de literatuur meerdere methoden bekend die worden gebruikt om in een marketingomgeving de wisselwerking tussen deelbeoordelingen en een eindoordeel weer te geven. Deze methoden zijn: Critical Incident Technique (CIT), Importance-Performance Analysis (IPA), logistieke regressie met dummyvariabelen, Penalty-Reward Contrast Analysis (PRCA), en analyse met behulp van Uniform Aggregation Operators (Uninorms).

We zoeken naar een methode die geen uitspraken over de data doet aan de hand van vooraf gedefinieerde verbanden tussen, of verhoudingen van attributen. De methodiek moet wel rekening houden met de verwachting van de klant met betrekking tot het product of de dienst (het “anchor”), en met het feit dat effecten als “Up-” of “Downward reinforcement” op kunnen treden. Vanhoof (2005) presenteert UniNorms als een methodiek die aan deze eisen voldoet.

Deze methode wordt hierna behandeld, waarna in hoofdstuk 8 op de voor-, en nadelen van alle technieken wordt ingegaan.

4 – Theorie UniNorms

Aggregatieoperatoren lenen zich uitstekend om meerdere scores samen te voegen tot één numerieke waarde. Een behoorlijk aantal families van aggregatieoperatoren is onderzocht, zie hiervoor Dubois (1985) [6]. Er is een familie van aggregatieoperatoren met voor dit onderzoek twee zeer belangrijke eigenschappen, te weten “full reinforcement” en compensatie. “Full reinforcement” is de term gebezigd voor het ondersteunen van zowel “downward reinforcement” als “upward reinforcement”.

De familie die deze eigenschappen beide heeft is de familie van uniforme aggregatieoperatoren, ook wel bekend als uninorms.

Een uninorm is formeel gezien een functie $U : [0,1] \times [0,1] \rightarrow [0,1]$ met de volgende eigenschappen:

beschrijving	eigenschap
$U(x, y) = U(y, x)$	Commutatief
$U(x, y) \leq U(u, y)$, als $x \leq u$	Monotoon
$U(x, U(y, z)) = U(U(x, y), z)$	Associatief
$\exists! \nu \in [0,1] \forall x \in [0,1] : U(x, \nu) = x$	ν is het neutrale element

Tabel 4.1

Verondersteld wordt dat de aggregatieoperator indifferent is met betrekking tot de volgorde van de parameters, wat er op uitkomt dat het voor de uninorm niet uitmaakt in welke volgorde de waarderingen van de deelaspecten staan. Commutativiteit betekent in dit verband dat de waarde van de uninorm hetzelfde is, wanneer een identieke set deelwaarderingen in een andere volgorde wordt verwerkt in de uninorm. Deze eis brengt complicaties met zich mee, zo betekent dit dat de aggregatieoperator bestaat uit ongewogen variabelen.

Het is niet mogelijk om van te voren meer of minder gewicht toe te schrijven aan een van de aspecten van een product, alle aspecten wegen even zwaar mee in het eindoordeel. Of dit een wenselijke eigenschap is zal in hoofdstuk 9 worden besproken.

De tweede eis die door Dombi (1982) (zie [5]) wordt gedefinieerd is de eis van monotoniteit van de uninorm. Gegeven een set waarden (deelwaarderingen) leidt het verhogen van een van de waarden nooit tot een verlaging van de waarde van de uninorm. Dit geldt ook omgekeerd, dus wanneer er een waarde wordt verlaagd zal dit nooit leiden tot een verhoging van de waarde van de uninorm (eindwaardering). In zowel het geval van een verhoging van die ene waarde of een verlaging ervan is het overigens ook mogelijk dat de waarde van de uninorm gelijk blijft. De enige eis die hier gesteld wordt is dat de uitkomst van de uninorm niet daalt bij een verhoging respectievelijk stijgt bij een verlaging van een parameter. Geheel ten overvloede kan vermeld worden dat dit in alle gevallen ook geldt voor het simultaan verhogen danwel verlagen van meerdere parameters.

Een uninorm moet ook voldoen aan de derde eis, namelijk associativiteit. Dit betekent dat het voor de uitkomst van de aggregatieoperator niet uitmaakt welke attributen als eerst verwerkt worden. Het voorbeeld uit de tabel geeft aan dat, wanneer de deelwaarderingen y en z samen worden genomen in een aparte uninorm en deze uitkomst met x er bij als input voor een tweede uninorm genomen wordt, dit dezelfde uitkomst geeft als de deelwaarderingen x en y eerst samen worden genomen.

De laatste eis aan een uninorm is dat er een zogenaamde neutrale waarde “ ν ” moet bestaan zodat wanneer een van de deelwaarderingen van een klant deze waarde aanneemt, dit noch een positieve, noch een negatieve invloed heeft op de eindwaardering.

Deze neutrale waarde kan geïnterpreteerd worden als het niveau wat de klant als verwachtingswaarde heeft voor de prestaties van het product op alle aspecten. Deze neutrale waarde wordt bepaald aan de hand van alle deelbeoordelingen en het eindoordeel, en wordt dus verondersteld gelijk te zijn voor alle te beoordelen aspecten van een product of dienst. Met deze kennis is het niet moeilijk om in te zien dat dit enige nadelen kan hebben op het model, of dat het in ieder geval enige restricties oplegt aan de accuratesse van de benadering van de respons van een willekeurige klant. In hoofdstuk 9 zal verder op dit onderwerp worden ingegaan.

Een uninorm is een benadering van het eindoordeel van een klant, en is dus niet eenvoudig op te stellen. De uninorm kan opgesteld worden aan de hand van formule (1). $g(x)$ staat voor de zogenoemde generatorfunctie met als eigenschap $g : [0,1] \rightarrow [-\infty, \infty]$ waardoor de uninorm als volgt gedefinieerd kan worden.

$$U(x, y) = g^{-1}(g(x) + g(y)) \quad (1)$$

Vanhoof (2005) [15] stelt dat de generatorfunctie g aan de volgende eisen moet voldoen:

De functie is continu op het interval $[0,1]$.

De functie is strikt monotoon stijgend.

Verder moet gelden: $g(0) = -\infty$ en $g(1) = \infty$.

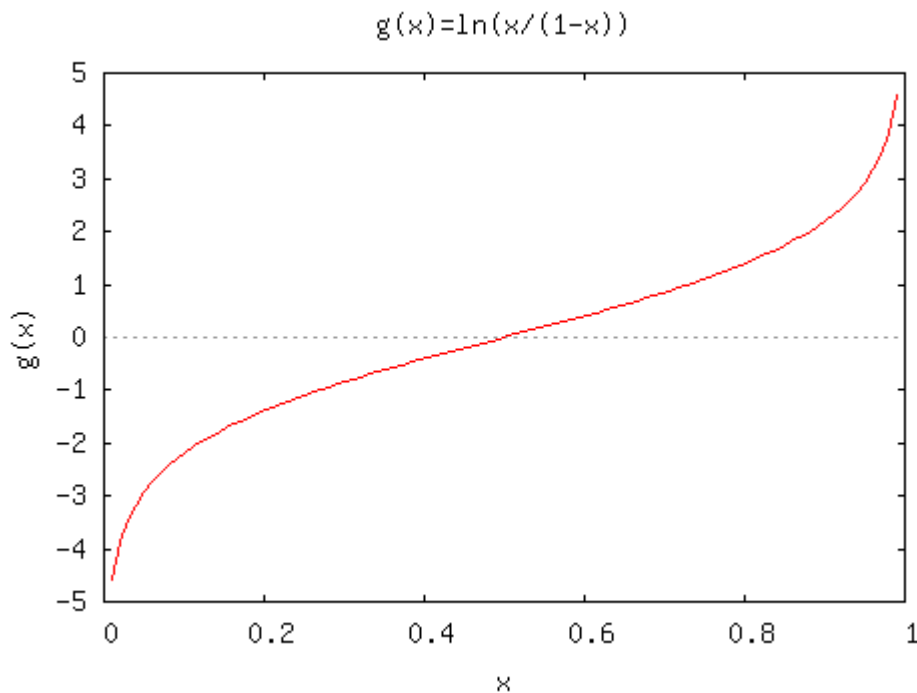
Dombi (1982) [5] geeft in zijn artikel als generatorfunctie $f(x) = \frac{e^x}{1+e^x}$. In dit onderzoek is $f(x)$ omgeschreven naar $g(x)$ door te stellen dat $g(x) = f^{-1}(x)$ waaruit volgt dat

$$g(x) = \ln\left(\frac{x}{1-x}\right), \quad (2)$$

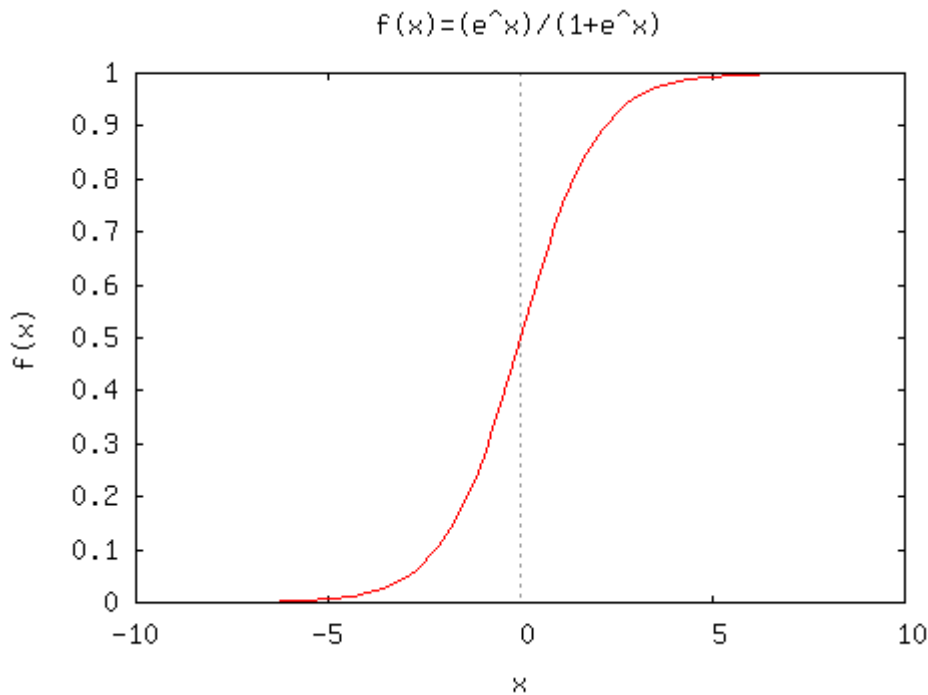
waardoor het nu voldoet aan de door Vanhoof gestelde eisen.

In figuur 4.1 is het functieverloop van $g(x) = \ln\left(\frac{x}{1-x}\right)$ weergegeven, terwijl figuur 4.2 $f(x) = \frac{e^x}{1+e^x}$ weergeeft.

Het is duidelijk te zien dat $g(v) = 0$ een bestaand punt is, terwijl de functie overal een stijgend verloop heeft.



Figuur 4.1



Figuur 4.2

Op basis van deze generatorfunctie $g(x)$ is het mogelijk om een hele familie van deze functies te bouwen, door de functie met α te verplaatsen.

Dombi (1982) toonde aan dat wanneer een functie $f(x)$ de generatorfunctie van de uninorm is, de functie $f_\alpha(x)$ met $\alpha: f(x+\alpha) = f_\alpha(x)$ dezelfde eigenschappen heeft als $f(x)$.

Toegepast op onze situatie levert dit ons $g_\alpha(x) = g(x) - \alpha$, doordat $g_\alpha^{-1}(x) = f_\alpha(x)$.

Wanneer deze formules toegepast moeten worden in programmatuur zoals tijdens dit onderzoek, zullen de formules moeten worden omgeschreven naar een bruikbare formule, waar ook enig rekenwerk mee te doen is. Zie hiervoor bijlage 11.1 – Implementatie.

Terugkomende op (2) en na verplaatsing van de functies met α komen we tot $g_\alpha(x) = \ln\left(\frac{x}{1-x}\right) - \alpha$.

(Met $f_\alpha(x) = \frac{e^{x+\alpha}}{1+e^{x+\alpha}}$)

Onze eerste definitie van een uninorm bij (1) is een functie van slechts 2 variabelen, te weten x en y . Aangezien we zelden datasets tegenkomen waarbij er twee variabelen zijn maar eigenlijk altijd over veel grotere aantallen spreken moet onze definitie aangepast worden.

Door de formule om te schrijven naar n variabelen met een sommatie zoals in (1) krijgen we

$$U(x_1, \dots, x_n) = g^{-1} \left(\sum_{i=1}^n g(x_i) \right). \quad (3)$$

Uitgaande van een dataset met n attributen waarbij de deelscores van een klant op de attributen $1, \dots, n$ worden weergegeven door s_1, \dots, s_n en de eindwaardering wordt weergegeven door S , komen we tot het volgende:

$$U(x_1, \dots, x_n) = U(s_1, \dots, s_n) = S \quad (4)$$

Deze formule (4) is zeer belangrijk, hier staat effectief dat de uninorm gelijk is aan de eindwaardering van een klant, wanneer de parameters van de uninorm gelijk zijn aan de deelwaarderingen van de klant over het betreffende product of over de betreffende dienst. Dit is de uninorm zoals benodigd is voor dit onderzoek, als een benadering van de eindwaardering gegeven de deelwaarderingen van de klant.

Gegeven de formules (3) en (4), en na verplaatsing met α geldt $U(s_1, \dots, s_n) = g_\alpha^{-1} \left(\sum_{i=1}^n g_\alpha(s_i) \right)$,

dan moet dus gelden:

$$g_\alpha(S) = \sum_{i=1}^n g_\alpha(s_i).$$

Hieruit kunnen we α als volgt oplossen:

$$g(S) - \alpha = \sum_{i=1}^n g(s_i) - n\alpha, \text{ ofwel}$$

$$(n-1)\alpha = \sum_{i=1}^n g(s_i) - g(S).$$

$$\alpha = \frac{1}{n-1} \left(\sum_{i=1}^n g(s_i) - g(S) \right) \quad (5)$$

Deze α is ook wel “displacement” genoemd, hiermee is het mogelijk de neutrale waarde te berekenen via

$$\alpha = g(v). \quad (6)$$

Dombi (1982) [5] heeft dit bij zijn “Theorem 3” bewezen, en zo komen we tot

$$v = g^{-1}(\alpha) = f(\alpha). \quad (7)$$

Deze neutrale waarde wordt verondersteld om de uitgangswaarde van het individu te zijn, deze waarde is in principe ook voor ieder individu anders of anders opgebouwd.

Ieder individu heeft een eigen neutrale waarde ten aanzien van het te beoordelen product of de te beoordelen dienst. Dit is, zoals al eerder genoemd, de verwachte score van de prestatie van het product of

de dienst als geheel. De hier behandelde theorie gaat er van uit dat deze neutrale waarde ook geldt voor ieder deelaspect, waarbij deze waarde ook gelijk blijft. Wanneer een klant dus verwacht een score van 5 te zien (op een schaal van 1 tot 10) op het product als geheel, is de verwachte waarde van ieder aspect van het product ook 5. Deze neutrale waarde is voor iedere klant anders of anders opgebouwd.

In de onderstaande tabel is tabel 3.1 uitgebreid met de berekende neutrale waarden.

Klant	s_1	s_2	s_3	S	ν
A	5	3	6	4	5.0
B	5	6	5	7	4.4
C	3	3	4	2	4.1
D1	5	7	6	6	6.0
D2	5	7	6	5	6.5

Tabel 4.2

Aan de hand van deze bovenstaande gegevens is het mogelijk de individuele uninorm vast te stellen. Gegeven het feit dat deze ‘aangeleerde’ uninorm dezelfde eigenschappen heeft als een generieke uninorm is het mogelijk om per attribuut te berekenen wat de deelscore voor invloed heeft op het gegeven eindoordeel S .

De individuele uninorm U_i heeft de volgende eigenschappen:

- U_i is associatief
- U_i is commutatief
- U_i is continu, behalve in de punten $(0,1)$ en $(1,0)$
- U_i is strikt monotoon
- $U_i(x, y, \nu) = U_i(x, y)$

Deze laatste regel geeft aan dat, wanneer men een individuele uninorm neemt en er een extra attribuut aan toevoegt wat dezelfde waarde als de neutrale waarde heeft, deze nieuwe individuele uninorm dezelfde eigenschappen en neutrale waarde heeft. Het toevoegen van een extra attribuut met als waarde de neutrale waarde van de uninorm heeft geen invloed. De nieuw geconstrueerde en de oorspronkelijke individuele uninorm zijn zelfs gelijk.

Om te zien wat een bepaald attribuut voor invloed heeft op de eindwaardering van een product is de “contribution” gedefinieerd. Op basis van de laatste drie genoemde eigenschappen kan formule (8) worden opgesteld, deze formule handelt over attribuut j van individu i .

$$\text{Contrib}(i, j) = U_i(s_1, \dots, s_{j-1}, s_j, s_{j+1}, \dots, s_n) - U_i(s_1, \dots, s_{j-1}, \nu_i, s_{j+1}, \dots, s_n) \quad (8)$$

In deze definitie van $\text{Contrib}(i, j)$ is het neutrale element ν weergegeven als ν_i om aan te geven dat dit het specifieke neutrale element van het betreffende individu is. Verder is gekozen voor de notering van de deeloordelen van een individu als s_1, \dots, s_n in plaats van $s_{i,1}, \dots, s_{i,n}$ omwille van de eenvoud van de notatie. Deze formule handelt wel degelijk over de deelscores per attribuut en per individu.

Wanneer ν bekend is kan de “contribution” (bijdrage) per attribuut berekend worden.

Door in de al bekende uninorm het gewenste attribuut te vervangen door het neutrale element ν wordt de invloed van dit attribuut vervangen door een invloed van 0. Door het verschil te nemen tussen de originele en deze aangepaste uninorm kan de invloed van het ene attribuut berekend worden op de waarde van de uninorm.

Klant	s_1	$\text{Contrib}(s_1)$	s_2	$\text{Contrib}(s_2)$	s_3	$\text{Contrib}(s_3)$	S	ν
A	5	+0.0	3	-2.0	6	+1.0	4	5.0
B	5	+0.5	6	+1.4	5	+0.5	7	4.4
C	3	-0.9	3	-0.9	4	-0.1	2	4.1
D1	5	-1.0	7	+1.0	6	+0.0	6	6.0
D2	5	-1.5	7	+0.6	6	-0.6	5	6.5

Tabel 4.3

Tabel 4.3 is een uitgebreide versie van tabel 4.2, waarbij de bijdragen per attribuut en per klant zijn opgenomen.

Wanneer we goed naar de bijdragen kijken is bij klant A mooi zichtbaar dat de positieve en negatieve bijdragen elkaar compenseren. De bijdragen zijn positief danwel negatief doordat de scores op de attributen respectievelijk hoger en lager zijn dan verwacht. Bij s_1 is de bijdrage nul omdat de score gelijk is aan de verwachte waarde, de neutrale waarde.

De beoordeling van klant B is een goed voorbeeld van “positive reinforcement”, de scores op de deelaspecten zijn hoger dan verwacht, de bijdragen zijn allemaal positief en de eindwaardering is de hoogste waardering van de klant. Het omgekeerde geldt voor klant C, dit fenomeen heet “negative reinforcement”. De scores op de deelaspecten zijn lager dan verwacht, de bijdragen zijn allemaal negatief en de eindwaardering is de laagste score die klant C heeft gegeven.

Om de kracht van de uninorm maar weer eens te laten zien zijn klanten D1 en D2 opgenomen, met ieder een andere eindwaardering maar dezelfde waarderingen op de aspecten zijn voor beide klanten zowel de verwachtingswaarde als (automatisch) de bijdragen verschillend van elkaar.

Nu zeggen deze individuele getallen niet zoveel over de populatie als geheel maar slechts over de geïnterpreteerde beoordelingen van de klanten.

Als over alle attributen j van individuen i de bijdragen zijn berekend kunnen deze geaggregeerd worden als een gewogen gemiddelde. Dit wordt per attribuutwaarde, per attribuut gedaan zodat er een overzicht komt van de gemiddelde bijdrage per attribuutwaarde, per attribuut. Deze gemiddelde bijdrage kan zowel

positief als negatief zijn, al naar gelang het attribuut. De verwachting is, doordat de uninorm strikt stijgend is, dat de bijdragen van de lagere waarden van de attributen negatief zijn, terwijl de hogere waarden van de attributen een positieve bijdrage op zullen leveren.

Dit ongewogen gemiddelde wordt gedefinieerd als impact, volgens formule (9).

$$\text{Impact}(j, x) = \frac{\sum_{\{i|s_j=x\}}^n \text{Contrib}(i, j)}{\#\{i | s_j = x\}} \quad (9)$$

NB: de sommatie van de “contributions” gaat over die individuen i met $s_j = x$.

Het resultaat van deze formule zal, gegeven een dataset zoals eerder genoemd, bestaan uit gemiddelde “contributions”, uitgesplitst per waarde van een attribuut.

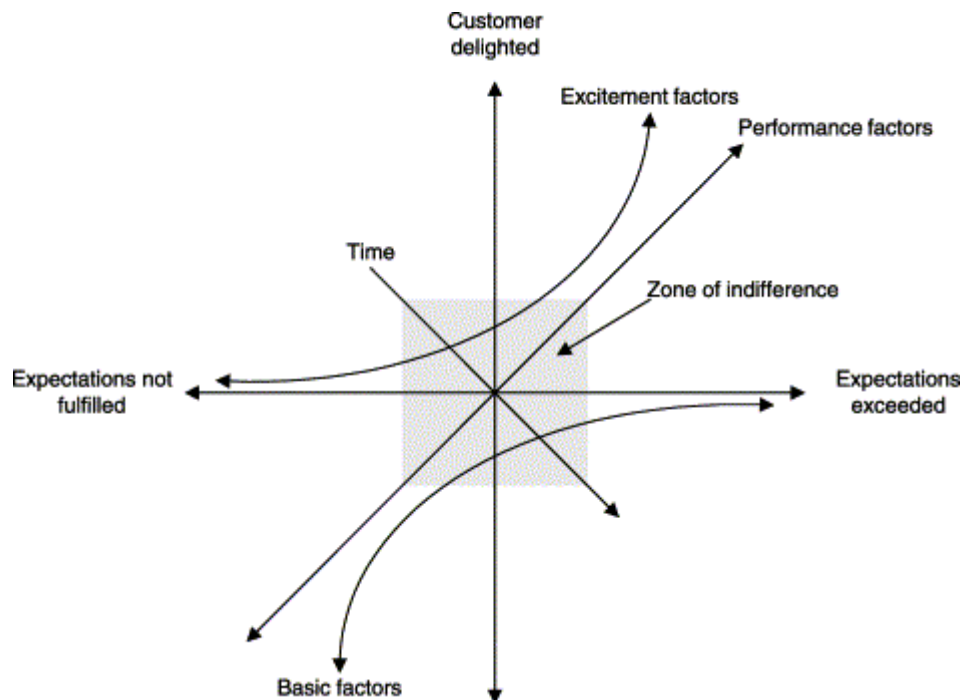
Wanneer een dataset j attributen heeft ($1 \leq j \leq 10$), en ieder attribuut heeft x verschillende waarden met x tussen 1 en 5, zal de uitkomst aan de hand van formule (9) een matrix zijn van 10 rijen en 5 kolommen.

Om dit inzichtelijk te maken kan deze matrix gevisualiseerd worden als een assenstelsel met daarin j lijnen met ieder x punten. Aan het verloop van een dergelijke lijn die ook wel “impact”-lijn wordt genoemd is af te lezen onder welke van de drie door Matzler, et al. (2004) (zie [9]) gedefinieerde soorten attributen het betreffende attribuut valt.

Matzler, et al. (2004) hebben de theorie van N. Kano in het artikel “Attractive quality and must be quality” uit “Hinshitsu” volume 14 (1984)¹ grondig bestudeerd en aangepast. Kano stelde een model voor waarin de invloed van een attribuut op het eindoordeel kan worden onderverdeeld in drie groepen. Deze drie verschillende typen zijn “Excitement factors”, “Performance factors” en “Basic factors”, weergegeven zoals in het onderstaande figuur.

Een attribuut wordt geclassificeerd op basis van de invloed van het attribuut op de eindwaardering. Deze invloed kan zowel positief als negatief of zelfs neutraal zijn.

¹ Dit artikel is niet opgenomen in de literatuurlijst daar het alleen in de Japanse taal is gepubliceerd.



Figuur 4.3

Een “Basic factor” is een attribuut wat op het best een neutrale invloed heeft op de eindwaardering. Wanneer een dergelijk attribuut een minder dan maximale score heeft gekregen van de klant zal deze klant niet tevreden (genoeg) zijn, en een negatieve(-re) waardering toekennen aan het product als geheel. Een attribuut in deze categorie zal door de klant als minimumeis worden gesteld waarbij het uitblijven van een goede prestatie een naar verhouding zeer negatieve invloed uitoefent op het eindoordeel.

Wanneer een attribuut valt onder de “Excitement factors” is het exact tegenovergestelde van toepassing. Wanneer een goede prestatie op een dergelijk attribuut uitblijft, zal er geen negatieve invloed zijn op het eindoordeel. Bij een goede of zeer goede prestatie zal de klant extra tevreden zijn over het product en dus een hoger oordeel vellen, een “Excitement factor” verrast de klant en genereert tevredenheid over de aanwezigheid.

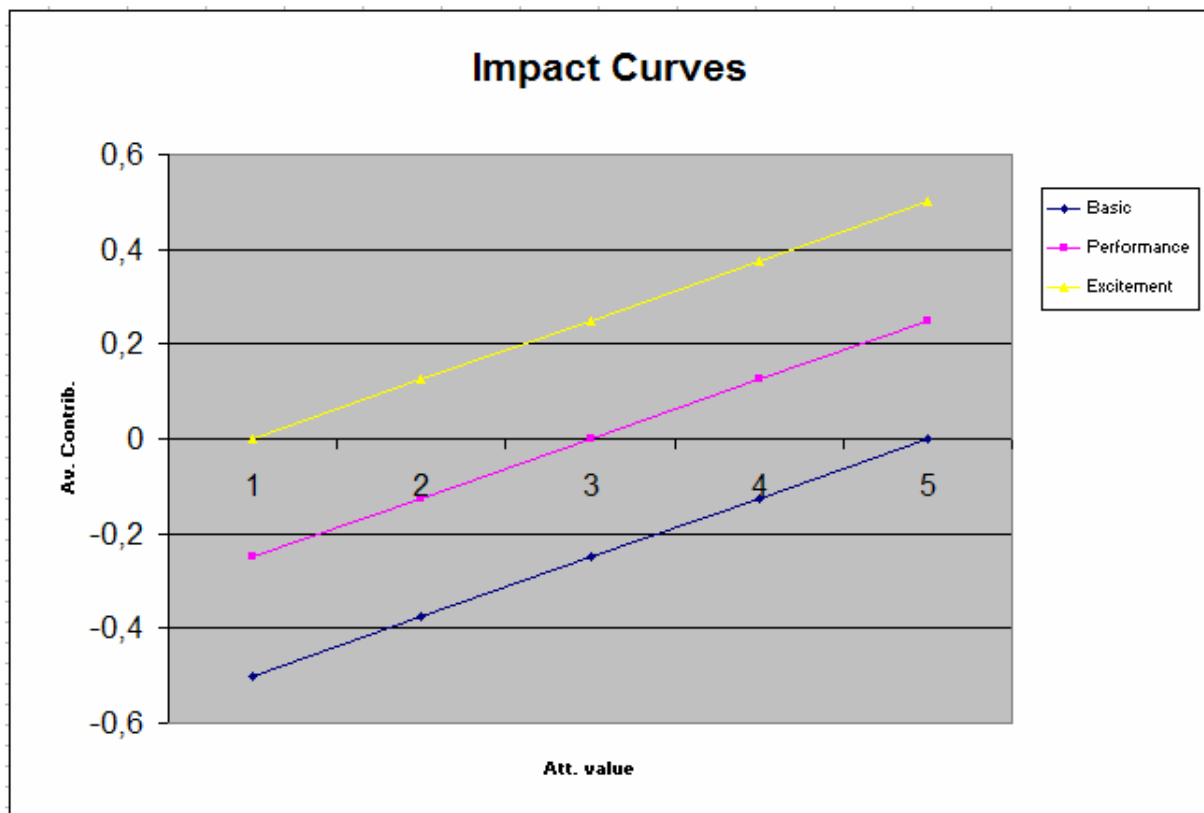
Een “Performance factor” hangt er qua eigenschappen een beetje tussenin, een negatieve prestatie levert een negatieve invloed op het eindoordeel op terwijl een positieve prestatie idealiter evenredig veel positieve invloed oplevert. In het bovenstaande figuur is de relatie tussen prestatie en invloed een lineaire symmetrische relatie.

De weergegeven “zone of indifference” (indifferentiezone) geeft aan dat een zeer kleine positieve of negatieve prestatie van een attribuut weinig tot geen invloed heeft op het eindoordeel. In de praktijk blijkt dat wanneer een attribuut binnen de indifferentiezone valt qua prestatie de invloed tenietgedaan wordt door een ander attribuut wat ook in dezelfde zone valt. Dit vanwege het feit dat er normaliter een flink aantal attributen in een onderzoek voorkomen en deze dus bij elkaar een dempende werking hebben.

Hieronder is een figuur weergegeven met de door de in dit onderzoek gebruikte methodiek gegenereerde uitkomsten, wanneer deze zo optimaal mogelijk zijn.

Let wel, de weergegeven data is fictief en niet gestoeld op enig onderzoek. Dit is slechts een grafische representatie om aan te geven wat idealiter de uitkomsten zijn van het onderzoek.

Goed te zien is het verloop van de lijnen, deze lopen (nagenoeg) parallel aan elkaar. Waar het verschil tussen de curven in zit is duidelijk, de onderste curve heeft een totaal ander bereik dan de bovenste curve. De derde en middelste curve zit er tussen in.



Figuur 4.4

Terugkomend op de “three-factor theory” is te zien dat door de opzet van het onderzoek en de zeer beperkte hoeveelheid punten van de curven, er wel degelijk verschillen zijn tussen de originele theorie van dhr. Kano en de hier gepresenteerde versie. Inhoudelijk zijn beiden identiek, op het koersverloop van de curven na. Kano ging uit van een non-lineaire curve voor zowel de “basic” als de “excitement” factoren. Hier zijn deze weergegeven als lineair verband.

Het doel van deze studie is een soort “managementtool” te presenteren waarmee relatief eenvoudig de kritieke succesfactoren van een product of een dienst geïdentificeerd kunnen worden. Hierna volgt een beschrijving van het uitgevoerde onderzoek om tot deze “tool” te komen.

5 – Beschrijving onderzoek

Het onderzoek wat in deze scriptie wordt beschreven is een herhalend onderzoek, waarbij de uitkomsten van de methode beschreven door Vanhoof ([15]) wordt herhaald op een andere, maar qua opzet vergelijkbare dataset uit een ander deel van de economie. In plaats van een productevaluatie door klanten is een totaal verschillende dataset gebruikt.

Bij dit onderzoek is gekozen voor een marketingdataset van ‘productevaluaties’, ingevuld door studenten over een vak wat in het betreffende trimester werd gevolgd. In het hierop volgende hoofdstuk zal de oorsprong van de data, evenals de data zelf uitgebreid worden toegelicht.

De keuze is op deze dataset gevallen omdat deze zich uitstekend leent voor dit onderzoek.

Er zijn redelijk veel respondenten, wat de dataset groot genoeg maakt voor een goede aggregatie. Bij een te klein aantal kan het zijn dat de data vertekend is door een aantal extreme waarden in de dataset, en/of kan de dataset een onvolledig beeld geven door het ontbreken van een bepaalde numerieke waardering.

Voor analyse van de data met uninorms, en vooral voor classificatie met de genoemde “three-factor theory”, is het van groot belang dat er een gemiddelde kan worden genomen van iedere mogelijke te geven waardering op ieder te beoordelen aspect van het product of de dienst. Deze specifieke wens zorgt voor de eis dat de dataset groot genoeg is, iets wat geen probleem lijkt te zijn met deze dataset.

Ook is analyse van de college-evaluatie dataset aantrekkelijk omdat er nog nooit op een dergelijke manier naar de data is gekeken. Het doel is hier om de aspecten aan te geven waarop de prestatie verhoogd kan worden zodat de eindwaardering zo maximaal mogelijk vergroot wordt. Momenteel is de interpretatie van de dataset een redelijk beperkte, waarbij er wordt gekeken naar de prestatie van een aspect ten opzichte van andere colleges, en ten opzichte van historische data van het betreffende college. Met de hier gepresenteerde methodiek is interpretatie op een ander niveau mogelijk, zij het niet op een vergelijkend niveau met andere colleges of de genoemde historische data.

Aangezien de theorie over uninorms geen implementatie biedt, maar slechts referenties naar mogelijke implementaties is in dit onderzoek gekozen voor een implementatie met $g(x) = \ln\left(\frac{x}{1-x}\right)$, analoog aan formule (2). Zie voor een uitgebreide uitwerking bijlage “11.1 – Implementatie”.

Voor een (correcte) verwerking van de dataset heb ik een aantal keuzes moeten maken en kleine compromissen moeten sluiten.

Zo is het, gegeven de eisen gesteld aan een uninorm door Vanhoof (2005), noodzakelijk dat de gebruikte gegevens op het interval $[0,1]$ liggen. Helaas was dit niet het geval met de brondata en was er een voorbewerking nodig om de data op het juiste interval te krijgen, deze bewerking is een simpele deling van de brondata door de maximale waardering die gegeven kan worden.

$$s_{i_{nieuw}} = \frac{s_i}{MAX_s} \quad (10)$$

$s_{i_{nieuw}}$ is de term die staat voor de aangepaste waarde van het oordeel, terwijl MAX_s de maximale waarde van s weergeeft.

Een sterke beperkende factor in dit alles is dat de maximale waarde van de invoer in een uninorm nooit 1 en de minimale waarde van de invoer nooit 0 mag zijn, aangezien formule (2) in die gevallen een ongeldige waarde geeft. Nu is het probleem met de ondergrens in de praktijk slechts een theoretisch probleem aangezien de respondenten in bijna alle gevallen een waardering moeten geven op een schaal van 1 tot 5 of een schaal van 1 tot 10. Het waarderen van een aspect of het product al geheel kan derhalve nooit in een cijfer nul resulteren, aangezien deze keuze niet bestaat.

De bovengrens is daarentegen wel problematisch. Wanneer een respondent de maximale waardering geeft aan een aspect, zal de aangepaste waarde gelijk zijn aan 1, en een ongeldige uitvoer tot gevolg hebben. Ik heb dit opgelost door in de programmatuur niet te delen door de maximale waarde van een oordeel (MAX), maar door $(MAX+1)$.

Er is bewust gekozen voor deze waarde, deze levert de minste vertekening op van de data.

Toegepast op een dataset met 1000 rijen, die ieder bestaan uit 20 willekeurige waarden, is het verschil goed te zien aan de hand van de zogenoemde “impact-curves”.

In een impact-curve is ieder punt de gemiddelde bijdrage van het betreffende attribuut bij die waarde aan het eindoordeel van de klanten. Een curve bestaat dus uit evenveel punten als dat er mogelijke waarden zijn die de waardering voor het attribuut kan aannemen. In onderstaande figuren zijn er een twintigtal curven weergegeven met ieder 5 waarden. Deze waarden zijn uitgezet op de horizontale as. “Att. Value” staat hier voor de waarde van de beoordeling van het attribuut. Op de verticale as is de gemiddelde bijdrage uitgezet, hier weergegeven als “Av. Contrib.”.

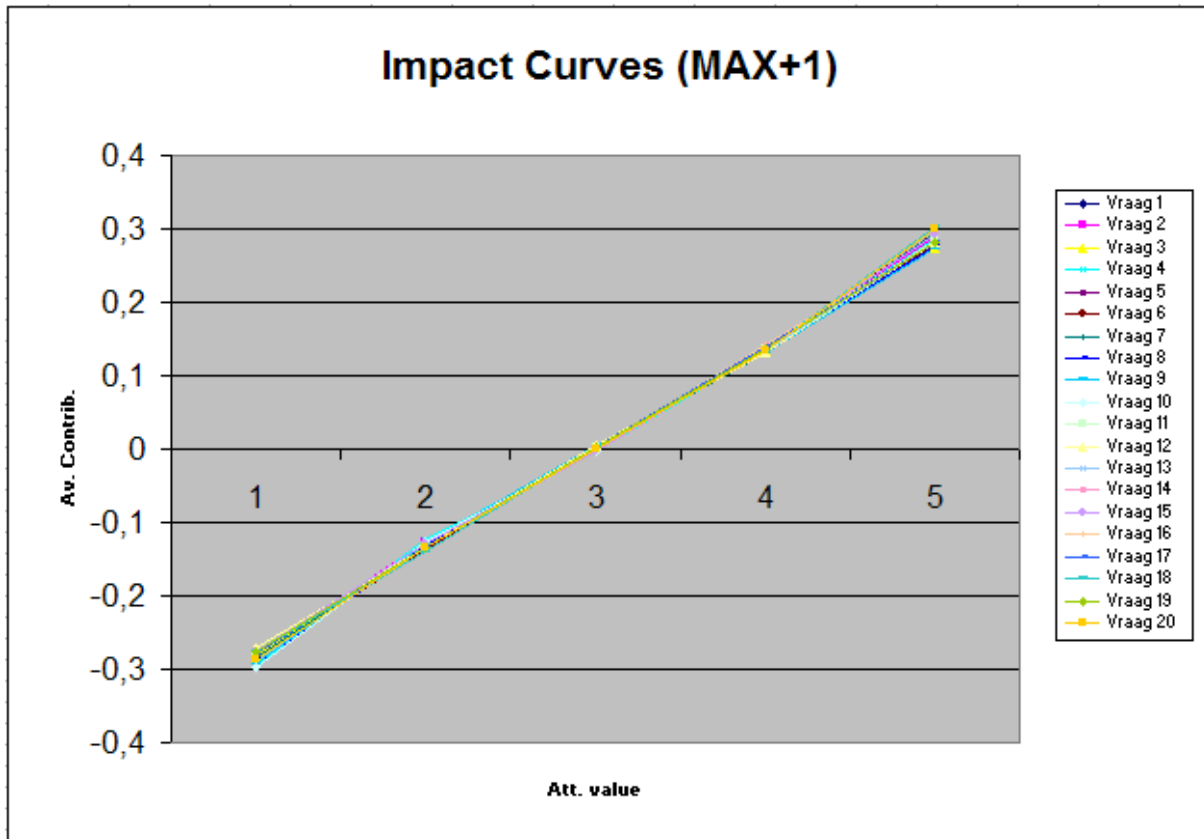
In beide figuren is de deling uit formule (10) aangepast tot

$$s_{i_{nieuw}} = \frac{s_i}{(MAX_s + \beta)}, \quad (11)$$

waarbij figuur 5.1 een β heeft van 1 en figuur 5.2 een β van 0.1.

Zoals logischerwijs volgt uit de eisen aan een uninorm zijn ook de hieronder weergegeven curven strikt stijgend. De willekeurige getallen die in deze figuren als brondata is gebruikt hebben een interessant effect op de curven, ze lopen niet alleen parallel maar zijn ook nog eens bijna identiek aan elkaar. Verder is ook

de plaatsing van de curven op de verticale as analoog aan de verwachtingen, zij het licht afwijkend. De verwachting is dat de absolute waarde van de minimale bijdrage even groot is als de maximale bijdrage, dat wordt in figuur 5.1 bevestigd. De gemiddelde bijdragen liggen hier bij benadering tussen -0.3 en 0.3. Eventuele afwijkingen zijn te wijten aan de willekeurige aard van de brongegevens en het relatief kleine aantal waarnemingen, de verwachting is dat er bij een zeer grote dataset geen vergelijkbare afwijkingen zullen optreden.

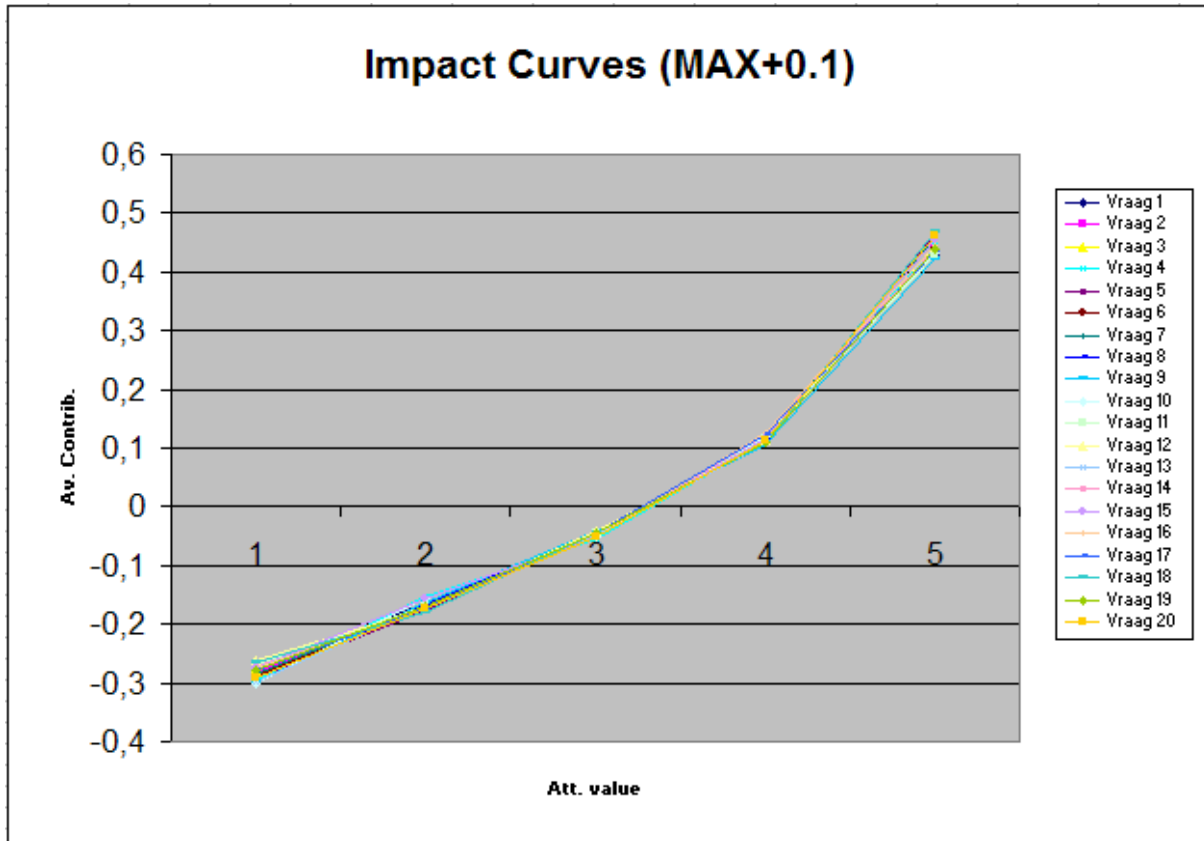


Figuur 5.1

Figuur 5.2 is afwijkend ten opzichte van figuur 5.1, dit heeft te maken met het kiezen van het getal β in (11). Doordat deze klein is gekozen heeft dit in formule (16) een groot effect op de bijdrage, het

gemiddelde komt hoger te liggen doordat $\lim_{x \rightarrow 1} \ln\left(\frac{x}{1-x}\right) = \infty$.

Hoe kleiner β , hoe groter de invloed op de impact-curven.



Figuur 5.2

Een tweede methodiek om de data op het interval $[0,1]$ te krijgen is het aanpassen van alleen die beoordelingen waar de respondent een maximale score heeft toegekend, naast de eenvoudige deling van (10).

De aangepaste deling van formule (11) is hieronder weergegeven, het verschil met de hiervoor genoemde formule is dat deze formule conditioneel is.

$$s_{i_{nieuw}} = \frac{s_i}{(MAX_s + \beta)} \text{ met } \beta = 0 | s_i \neq MAX_s \quad (12)$$

Een probleem met de gebruikte programmatuur is dat een respondent slechts scores kan geven op een schaal waar alleen gehele getallen op voorkomen. Natuurlijk is het mogelijk om deze schaal te vergroten of te verkleinen, zolang de brondata en de gebruikte schaal overeenkomen is hier geen probleem. Het aanpassen van alleen de maximale scores is lastiger, aangezien deze alleen een stapje omlaag naar het volgende gehele getal gezet kunnen worden. In het geval van het gebruikte voorbeeld, en ook de te gebruiken datasets in het onderzoek, zijn de gegeven beoordelingen gehele getallen tussen 1 en 5. Aanpassing van alle scores met de waarde 5 zou resulteren in een verlaging van die beoordeling naar het cijfer 4, wat de dataset ‘smaller’ maakt, er zijn minder verschillende beoordelingswaardes. Dit beïnvloedt

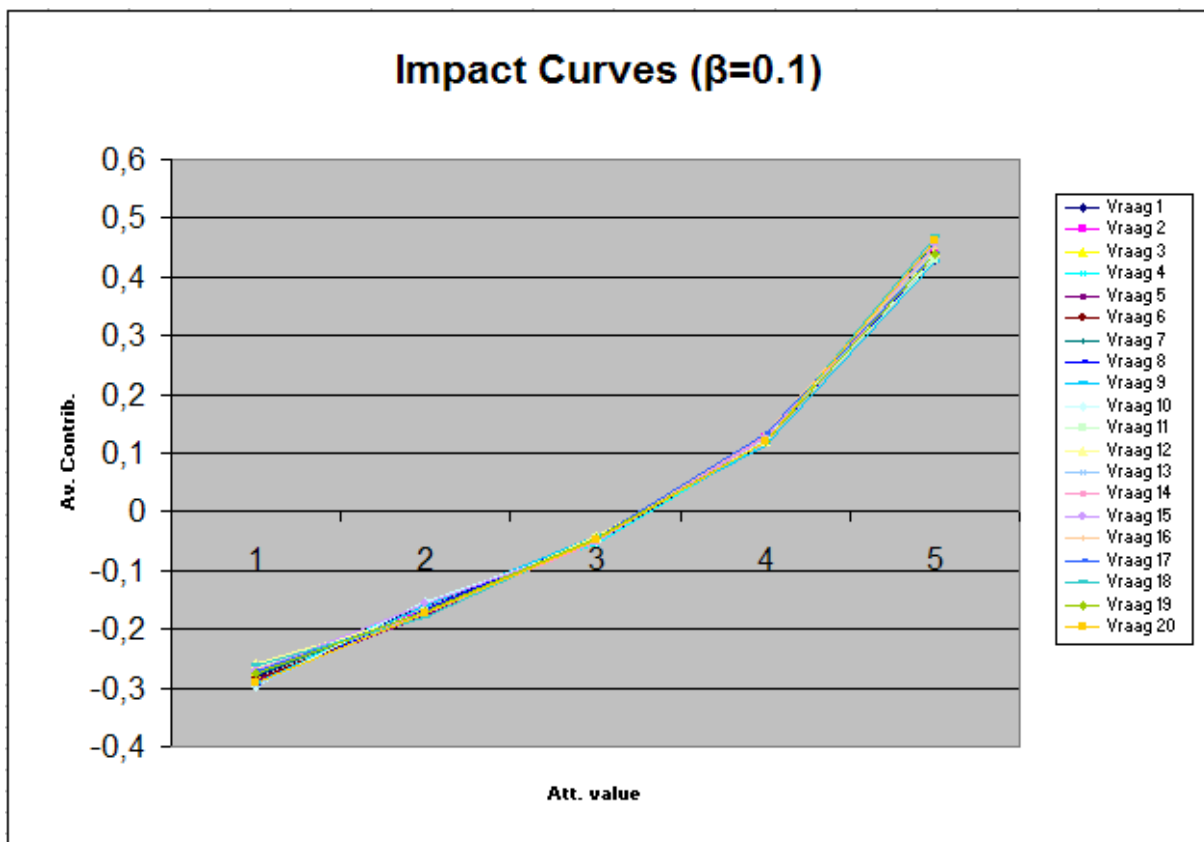
alle eigenschappen van de dataset, waaronder het gemiddelde, de mediaan, de modus en het belangrijkste van allemaal: de onderlinge verhoudingen in de dataset zelf.

Voordelen van deze methodiek zijn er wel degelijk, de dataset hoeft minder ingrijpend veranderd te worden. Slechts de maximale waarderingen worden wezenlijk veranderd, en de rest van de waarden wordt alleen gedeeld door het maximum.

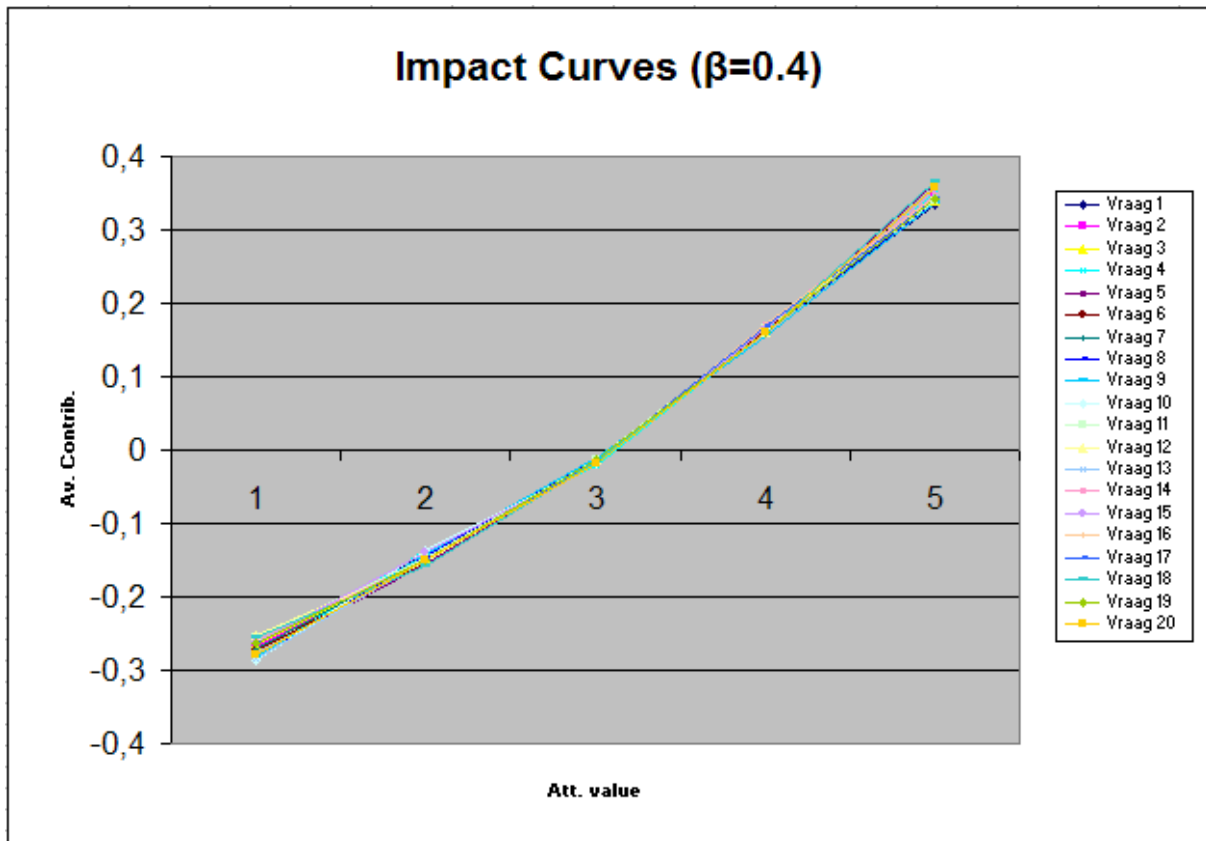
Bij een verandering van de verhouding van de data bestaat de kans dat de dataset dergelijke kleine onderlinge verschillen krijgt dat deze onbruikbaar is geworden. Om dit risico niet te lopen is er voor gekozen om de hier genoemde methode niet te gebruiken.

Om de nadelen van deze methode te illustreren zijn hieronder de gegenereerde “impact curves” weergegeven voor verschillende waarden van β . De keuze voor een bèta van 0.4 klinkt misschien verrassend, maar in werkelijkheid is dit een van de laatste waarden van bèta voordat de aan te passen beoordeling een klasse omlaag schuift. Bij $\beta = 0.5$ wordt de beoordeling afgerond van het maximale getal naar $MAX_s - 1$.

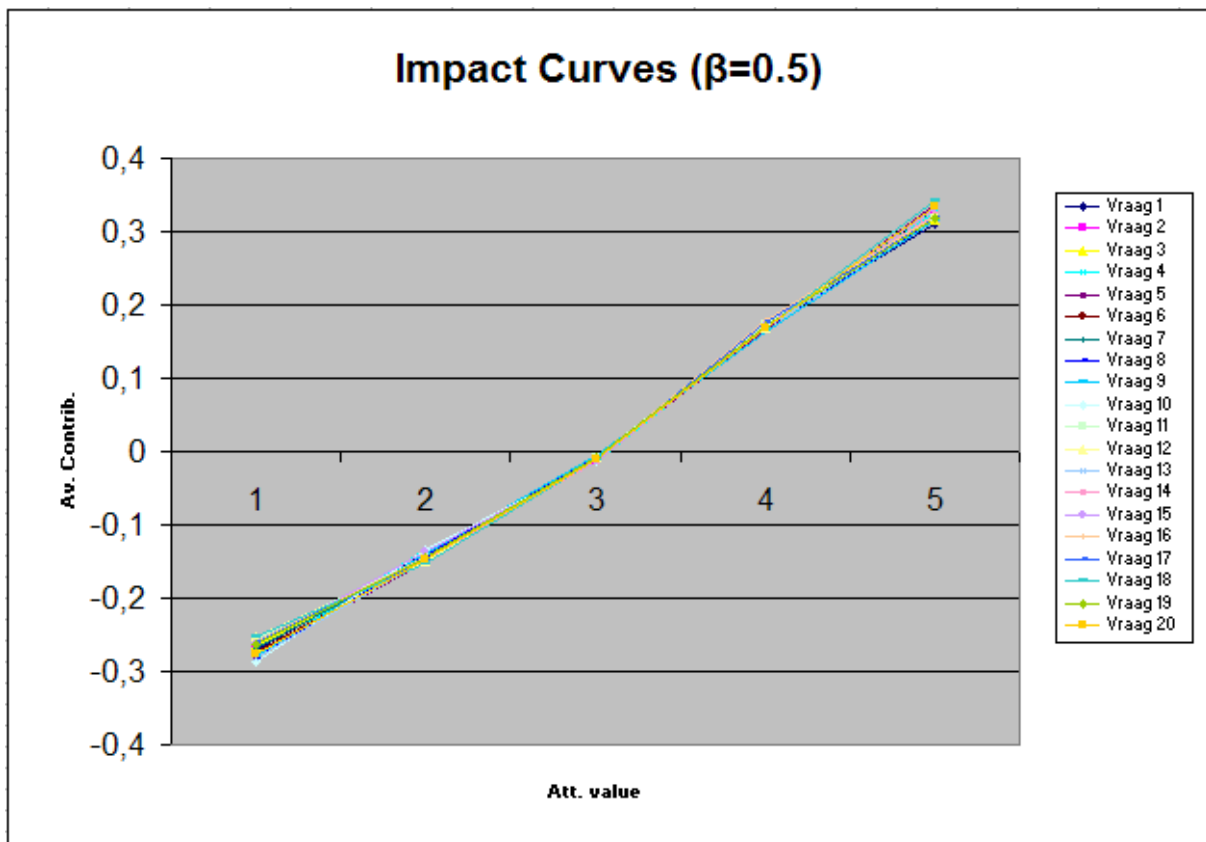
Ter referentie is ook $\beta = 0.1$ opgenomen om de abrupte stijging van de curven te laten zien die veroorzaakt wordt door de genoemde aanpassing.



Figuur 5.3



Figuur 5.4



Figuur 5.5

Mede door het veranderde verloop van de curven en het significante effect die deze koersverandering heeft op een eventuele interpretatie is er voor gekozen om deze methode niet toe te passen, en te kiezen voor de eerstgenoemde methode om de data op het juiste interval voor verwerking te krijgen.

Een ander probleem heeft te maken met de grootte van de gebruikte datasets. Vanwege het relatief kleine aantal correcte en volledige reacties van de respondenten is de dataset niet compleet voor een volledige interpretatie met behulp van de genoemde “three factor theory”. Voor het juist weergeven van de geïnterpreteerde data in een “impact curve” is een volledige dekking van alle mogelijke waarden van de beoordelingen van alle aspecten benodigd. Anders gezegd, alle te geven waarderingen moeten voor alle beoordelingen in de dataset aanwezig zijn, wanneer er 20 te geven waarderingen zijn die ieder 5 mogelijke waarden aannemen zijn er dus in ieder geval $20 \cdot 5$ verschillende waarderingen nodig. Voor een gemiddelde als de gemiddelde bijdrage die benodigd is voor een “impact curve” is het idealiter natuurlijk niet 1 waarde, maar zijn het meerdere waarnemingen die een punt in een dergelijke curve definiëren.

Meer over de grootte van de gebruikte datasets in het volgende hoofdstuk.

Om de toch al niet al te substantiële datasets niet te veel te beïnvloeden is er voor gekozen om de onvolledige “impact curves” als zodanig te behouden, en deze niet buiten beschouwing te laten. Weglating zou significante invloed op de interpretatie kunnen hebben, vandaar dat de waarderingen op alle aspecten intact zijn gelaten ook al geeft dit een ietwat vreemd beeld op de extreme waarderingen. Vooral de lagere waarderingen lijken hier onder te lijden, de respondenten hebben veelvuldig niet zeer negatieve beoordelingen gegeven, zodoende zijn de waarderingen “1” en “2” ondervertegenwoordigd. Slechts een enkele keer is de hoogste score niet toegekend.

Deze afwezigheid van beoordelingen is terug te vinden in de grafieken aan de anomalieën, de niet-bestaande waarden zijn weergegeven als 0, waardoor de curve een ietwat curieus uiterlijk krijgt met de extra knik.

Idealiter ziet een “impact curve” er uit als een van de curven uit figuur 5.2, maar helaas is dit niet altijd het geval, getuige de curven in de later volgende bespreking van de resultaten.

In het hierop volgende deel zullen de gebruikte datasets tot in detail besproken worden.

6 - Beschrijving data

6.1 - Oorsprong data

De data die in het onderzoek is gebruikt bestaat uit een drietal verschillende datasets, allemaal met dezelfde oorsprong.

Alle datasets zijn afkomstig van de Rotterdam School of Management (RSM), specifiek van de Nederlandse opleidingen die daar aangeboden worden.

Deze datasets zijn gegevens uit door studenten ingevulde evaluatieformulieren over een drietal verschillende vakken in dezelfde tijdperiode, te weten het collegejaar 2004-2005. De data wordt verzameld in een applicatie ontwikkeld door een externe partij, maar volledig in beheer van RSM. De eindverantwoordelijke voor deze applicatie zorgt er voor dat de vragenlijsten ingevuld, geactiveerd en onder de studenten verspreid worden. De evaluatieformulieren werden in een eerder stadium wel schriftelijk verspreid, sinds de komst van de externe applicatie wordt alles digitaal afgehandeld. De betreffende applicatie heeft zelf ook beperkte interpretatiefuncties waarbij er op een relatief laag niveau naar de data gekeken kan worden.

De betreffende vakken worden aan een flink aantal studenten aangeboden, gemiddeld aan ongeveer 1000 studenten per vak. Uit privacyoverwegingen zullen noch de vakken, noch de betreffende docenten hier gepubliceerd worden. De brongegevens zijn ook tijdens het onderzoek volledig geanonimiseerd gebruikt, de persoonsdata hebben geen enkele toegevoegde waarde aan het onderzoek. De datasets zullen vanaf nu aangeduid worden met de letters A, B en C. Hieronder volgt een kwalitatieve en kwantitatieve beschrijving.

6.2 - Kwaliteit en kwantiteit

Zoals genoemd is de data afkomstig van evaluatieformulieren die studenten in konden vullen naar aanleiding van het volgen van een vak. Deze vragenlijsten bestaan in de regel uit een behoorlijk aantal vragen over inhoudelijke onderdelen van het vak, de docenten en het gebruikte lesmateriaal.

De vragenlijst wordt op twee verschillende momenten in twee aparte delen aan de student aangeboden, het overgrote deel (waaronder het eindoordeel) wordt aangeboden na alle colleges maar voor tentaminering, de resterende vragen worden na de tentaminering aangeboden.

Het aantal 'restvragen' is zeer beperkt, in de regel is dit slechts een drietal vragen over de toets, en of deze wel of niet aansluit bij de aangeboden leerstof. Aangezien deze vragen pas na het vullen van het eindoordeel worden gesteld, zijn deze dus ook niet meegenomen in het onderzoek. Ook zijn er een aantal open vragen opgenomen die niet gebruikt worden bij de interpretatie van de gegevens.

6.2.1 - Dataset A

De originele dataset bestaat uit een 27-tal vragen, waarvan er een drietal naderhand gesteld zou moeten worden. Er zijn ook drie open vragen aanwezig.

Het aantal vragen over onderdelen van de cursus is dan ook 20, er is ook een overall-vraag aanwezig. 201 respondenten hebben de vragenlijst ingevuld, van de 996 studenten aan wie de lijst is aangeboden. Nadere bestudering van de data gaf aan dat er een flink aantal studenten is die de lijst niet compleet hebben beantwoord. Er blijven 146 complete lijsten over die gebruikt gaan worden.

6.2.2 - Dataset B

Deze dataset heeft dezelfde opbouw als dataset A, 27 vragen, 3 open vragen, 3 niet gestelde vragen, 1 vraag over de cursus als geheel.

164 respondenten hebben de vragenlijst ingevuld, van de 862 studenten aan wie de lijst is aangeboden. Er blijven 122 complete lijsten over die gebruikt gaan worden.

6.2.3 - Dataset C

Deze dataset heeft een andere opzet dan de vorige twee, de vragenlijst bestaat uit 28 vragen, waarvan 3 open vragen, 1 vraag over de cursus als geheel, en geen enkele vraag die pas na de tentaminering wordt gesteld.

Hierdoor zijn er dus 24 deelvragen, in plaats van de gebruikelijke 20.

149 respondenten hebben de vragenlijst ingevuld, van de 1025 studenten aan wie de lijst is aangeboden. Er blijven 86 complete lijsten over die gebruikt gaan worden.

Hieronder een cijfermatig overzicht van de gegevens:

dataset	vragen	studenten	respondenten	percentage ²	compleet	percentage ³
A	20	996	201	20,2	146	72,6 (14,7)
B	20	862	164	19,0	122	74,4 (14,1)
C	24	1025	149	14,5	86	57,7 (8,4)

Tabel 6.1

² Het percentage respondenten ten opzichte van het totale aantal studenten.

³ Het percentage complete vragenlijsten van het aantal respondenten, met tussen haakjes het percentage complete vragenlijsten ten opzichte van het totaal aantal ingeschreven studenten voor het betreffende vak.

7 – Toepassing op casussen

Vanwege de zeer grote compactheid van de resultaten is een geplande interpretatie van de onderzoeksgegevens met behulp van het drie-factoren model zoals eerder behandeld discutabel geworden. Aan het einde van dit hoofdstuk zal dit verder besproken worden.

7.1 - Dataset A

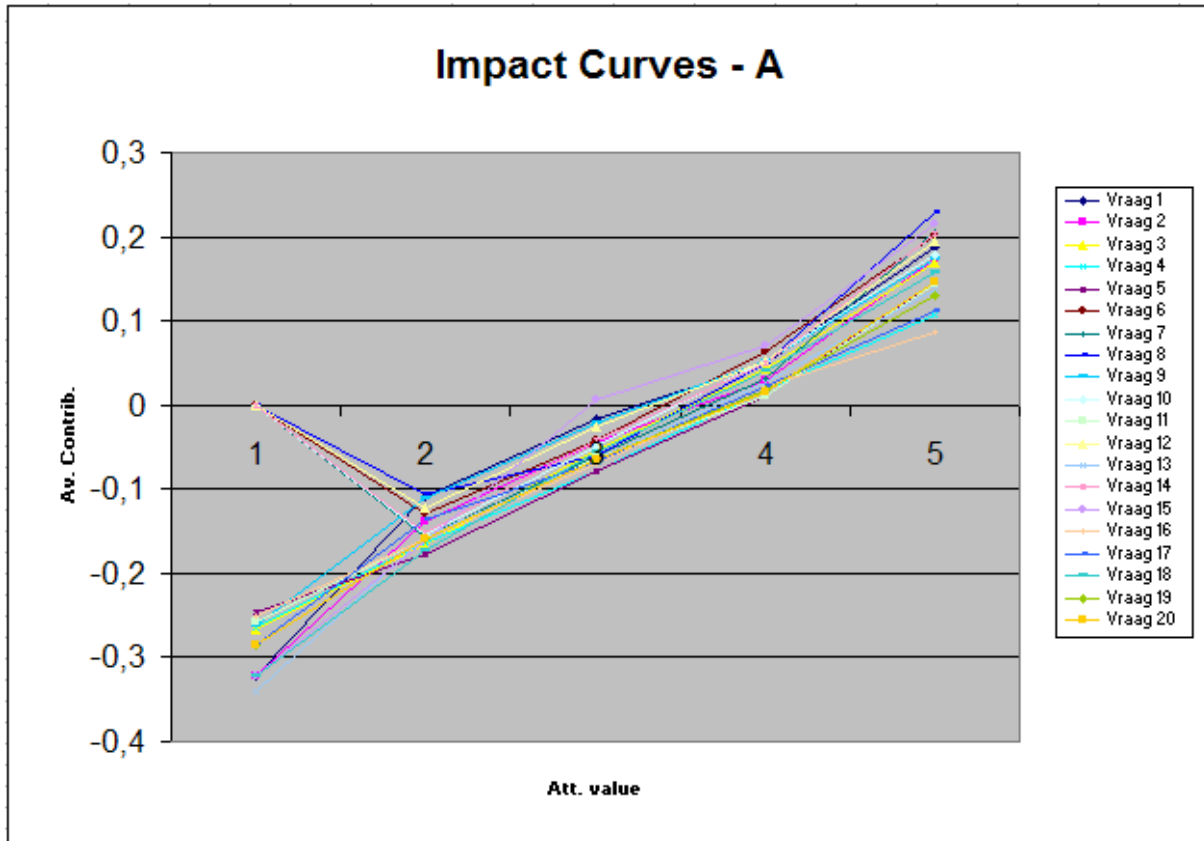
In figuur 7.1 is goed te zien dat de curven van deze dataset zeer compact rond de per definitie oninterpreteerbare impact-curven van de willekeurig gegenereerde brondata uit hoofdstuk 5 liggen.

Een van de weinige volledig valide uitspraken die over deze dataset gedaan kan worden op basis van onderstaande gegevens is het feit dat de studenten een gemiddelde verwachting hadden die ergens tussen een rapportcijfer 3 en 4 in lag. Hier snijden de meeste curven namelijk de horizontale as, een teken dat de gemiddelde bijdrage positief wordt.

De spreiding van de bijdrage bij een maximale score is redelijk groot, zo heeft vraag 16 de laagste maximale bijdrage, terwijl vraag 8 de grootste maximale bijdrage heeft. Het verschil tussen de twee is ruim 160 procent, dit terwijl het procentuele verschil tussen de kleinste en grootste minimale bijdrage (respectievelijk vraag 5 en vraag 13) op 38 procent blijft steken.

Vraag 8 is een zeer algemene vraag, wat de bijdrage natuurlijk zeer positief beïnvloedt. Wanneer men positief antwoordt op een dergelijk algemene vraag zal het eindoordeel bijna zeker ook positief zijn, en vice versa. Een overzicht van de gestelde vragen in een geanonimiseerde versie is weergegeven in bijlage 11.3 - Datasets, waarbij 11.3.1 de vragenlijst van deze dataset is.

Wat verder in deze dataset opvalt, is het koersverloop van vraag 15. Het bijdragepatroon loopt van relatief zeer negatief tot relatief zeer positief, met een vreemd aandoende positieve impuls bij een gemiddelde prestatie. Deze vraag is evenals vraag 16 een vraag over de individuele prestaties van een professor, in dit geval de persoon die de meeste colleges verzorgt wat er voor zorgt dat deze persoon een waarschijnlijk significante invloed heeft op de tevredenheid van de studenten. Desalniettemin lijkt het er op dat deze uitkomst een anomalie is veroorzaakt door het kleine aantal respondenten en daardoor het verwaarloosbare aantal maximale en minimale beoordelingen. Deze kunnen een onevenredig grote invloed uitoefenen, iets wat hoogstwaarschijnlijk hier mooi grafisch is weergegeven. Bij een ruimer aantal waarnemingen ligt het in de lijn der verwachting dat de extremen zullen afzakken en het hoge gemiddelde bij een waardering van 3 wat zal afnemen.



Figuur 7.1

Terugkomend op de eerdere opmerkingen over de vreemde “knik” in de curven zoals bij de waarderingen van 1 in een zestal curven: deze worden veroorzaakt door afwezigheid van de data. Er zijn simpelweg geen studenten geweest die deze vragen een beoordeling van “1” hebben toegekend. Voor de volledigheid: het gaat hier om vragen 6, 7, 8, 10, 12 en 14.

Er is geen duidelijke reden voor de afwezigheid van dergelijke beoordelingen gebaseerd op de vragen zelf, hier is ook verder geen onderzoek naar gedaan.

Globaal gezien is ongeveer de helft van de hier beoordeelde aspecten te classificeren als “performance” factor. Slechts een klein aantal, te weten vragen 5 en 11 is vanwege de ligging te classificeren als “basic” factor waarbij vooral vraag 5 (“Have you seriously prepared for your workshops”) er niet echt bij hoort.

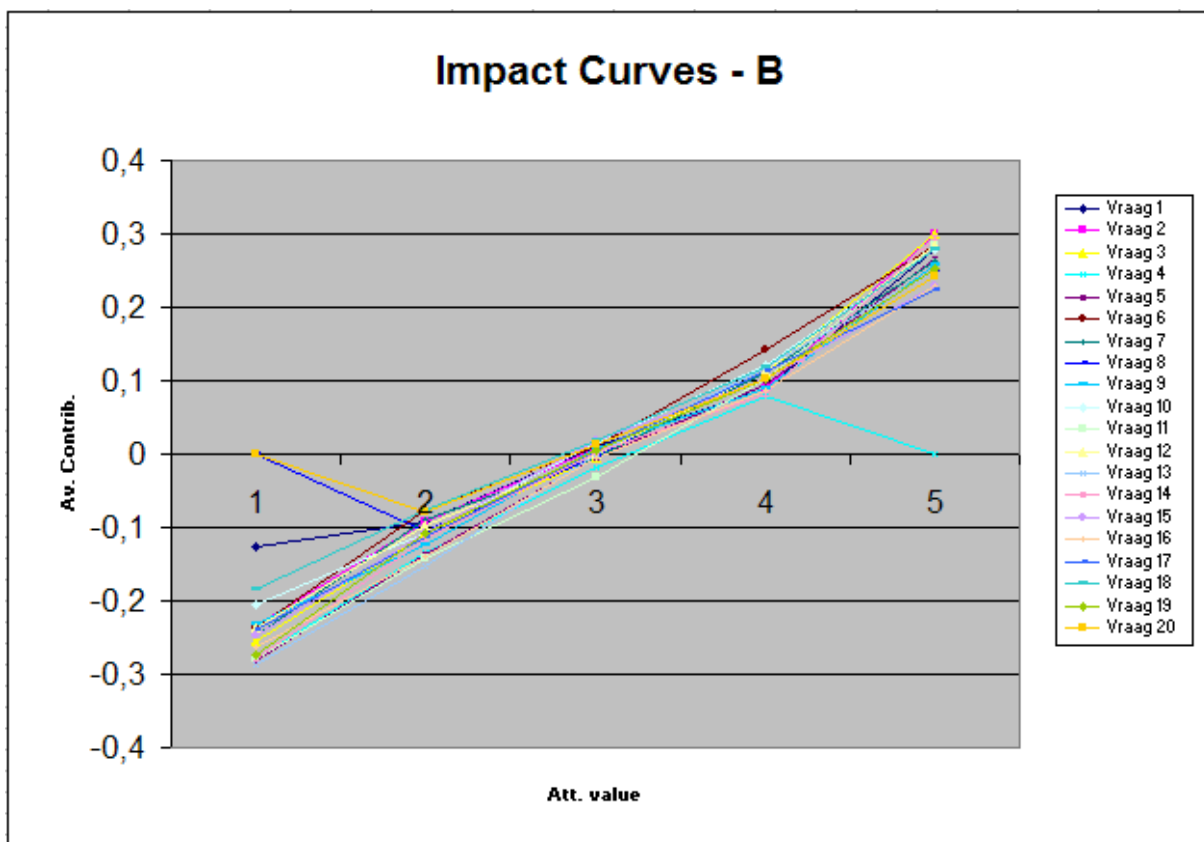
De andere vraag handelt over de stimulatie tot zelfstudie, en dit lijkt een meer valide uitkomst. Gebaseerd op deze uitkomst kan er als conclusie gezegd worden dat studenten graag gestimuleerd worden tot zelfstudie en niet alles voorgekauwd tijdens het college mee krijgen. Overigens is een daadwerkelijke valide conclusie niet het doel van dit onderzoek, eventuele aanbevelingen als bovenstaande zullen niet-steekhoudend zijn in een volledig uitgevoerd onderzoek onder de studenten.

7.2 – Dataset B

Ook voor deze dataset gelden zo goed als dezelfde beperkingen en opmerkingen over de set als geheel, het aantal respondenten is niet groot genoeg om anomalieën uit te sluiten, en er missen wat waarden in figuur 7.2.

Een groot verschil met de hiervoor behandelde dataset is het feit dat de gemiddelde curve een stuk hoger ligt, dit zou betekenen dat de studenten die deelnemen aan dit college een lagere verwachting hebben ten aanzien van dit vak. Deze conclusie komt overeen met het grotere bereik van de bijdragen, de maximale bijdrage is afgerond maar liefst 0.3, een kleine 30% hoger dan bij A.

De hier weergegeven impact-curven zijn zonder uitzondering te classificeren als “performance factor”, er is een nagenoeg even grote negatieve bijdrage bij een slechte prestatie als een positieve bijdrage bij een positieve prestatie. Verder is de bijdrage van vraag 1 (“What is your opinion on the overall organisation of this course”) bij een waardering van 1 af te doen als een anomalie gezien het feit dat deze bijdrage gebaseerd is op slechts 2 waarnemingen waarvan er 1 een behoorlijk afwijkende werking heeft op het gemiddelde. Wanneer deze beoordeling zou zijn weggelaten zou de bijdrage zijn afgenomen tot ongeveer -0.2 , iets wat beter strookt met de andere curven.



Figuur 7.2

7.3 – Dataset C

Dataset C heeft een groter aantal ontbrekende waarden, en is de enige dataset waar niet alleen de beoordeling “1” mist, maar ook een “2”. Gezien het feit dat deze vraag nummer 8 (“How difficult is the content of this course”) een behoorlijk goede representatie van de dataset dient te geven door het globale karakter van de vraag is dit een goede maatstaf voor alle beoordelingen van de respondenten. Uit de afwezigheid van beoordelingen uit de laagste twee categorieën is af te leiden dat men de cursus niet als te moeilijk heeft beoordeeld, wat strookt met het beeld dat ontstaat wanneer alle data in acht genomen wordt. De gemiddelde waardering voor welke er een positieve bijdrage geldt, is ook hier, analoog aan dataset A, ongeveer gelijk aan $3\frac{1}{2}$. Deze dataset heeft alleen een kleinere spreiding van de gegevens, rond de middelste classificatie zijn de curven gebundeld, alleen bij de extreme waarderingen (“1” en “5”) is er een spreiding zichtbaar. Vooral bij de hoge maximale waardering is de spreiding erg groot, het verschil tussen de laagste en hoogste is ruim 430 procent of 0.23 punt bijdrage. Dit terwijl de spreiding op de laagste maxima slechts op zo’n 63% of 0.13 punt bijdrage uitkomt.

De extra grote spreiding kan een tweetal oorzaken hebben, te weten de eerder besproken mogelijkheid van anomalieën in de dataset door het te kleine aantal waarnemingen maar ook de voorkeuren van de studenten. Het is mogelijk dat de gemiddelde student redelijk snel een hoge score toekent aan een aspect, maar nog niet geheel tevreden is met het eindresultaat. Dit kan de bijdragen van een aspect negatief beïnvloeden, iets wat vooral bij de lagere maxima op lijkt te treden.

De uitkomsten bij de drie vragen (respectievelijk vraag 21, 22 en 23) die een niet-strikt stijgend verloop kennen zijn per definitie incorrect. “Impact-curven” zijn strikt stijgend, en dus kan er geconcludeerd worden dat de uitkomsten die hier ten grondslag aan liggen anomalieën zijn die opgetreden zijn door het gebrek aan meerdere waarnemingen in de brondata, iets wat (helaas) al eerder is geconcludeerd.

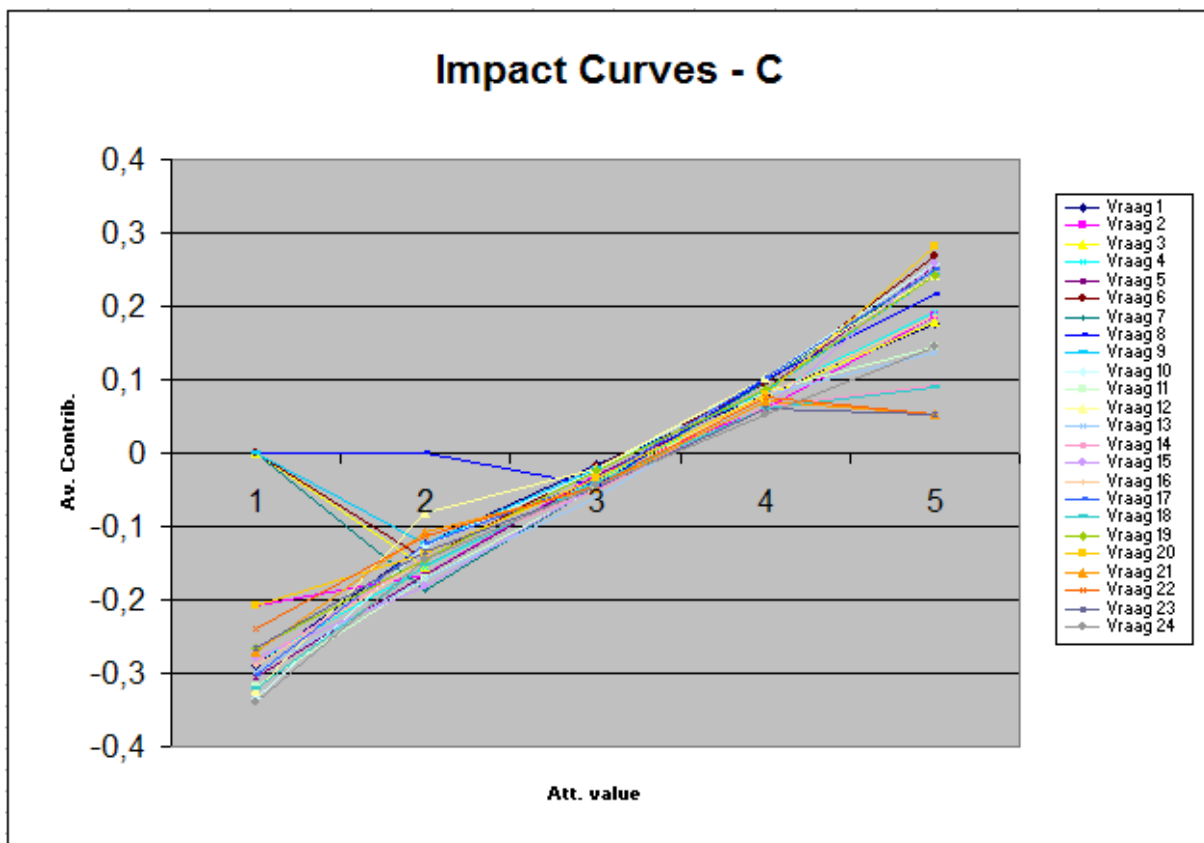
Opvallend aan deze drie vragen is dat ze allen een vergelijkbare opzet hebben. Vraag 21 luidt: “Wat is your opinion on the lectures of: Professor C”, terwijl vraag 22 handelt over professor D, en 23 handelt over professor E. Dit heeft geresulteerd in een vrijwel identieke score voor de drie vragen bij een waardering van “5”. Dit is een noviteit en waarschijnlijk puur toeval, hoewel het ook een heel andere reden kan hebben.

Gezien de lengte van de vragenlijst die de studenten in dienden te vullen, en het feit dat er 5 vragen zijn die dezelfde opzet hebben als de drie hierboven genoemde vragen kan het zijn dat de studenten ‘moe’ werden van het invullen en massaal dezelfde antwoorden hebben gegeven. Gezien de statistieken van de dataset is dit een correcte conclusie, de antwoorden van de studenten verschillen miniem tussen de vragen in kwestie. Zie hiervoor tabel 7.1 hieronder.

Waardering	Vraag 19	Vraag 20	Vraag 21	Vraag 22	Vraag 23
1	2	1	3	2	3
2	8	4	3	3	5
3	27	37	64	62	63
4	34	33	14	17	13
5	15	11	2	2	2

Tabel 7.1

Bovenstaande tabel geeft ook het numerieke probleem weer, terwijl er ruim 60 bekende waarderingen met een “3” zijn bij vragen 21, 22 en 23 zijn dat er slechts 3, 5 en 2 bij respectievelijk de waarderingen “1”, “2” en “5”. Uitwassen zijn hierdoor veel frequenter en kunnen een fout beeld geven.



Figuur 7.3

Classificatie met behulp van de drie-factor theorie is zoals eerder aangestipt een precare aangelegenheid geworden door het feit dat het onderscheid tussen de verschillende klassen onduidelijk is. In de gevallen van de hier bovenstaande datasets heeft dit hoogstwaarschijnlijk te maken met het feit dat er veel compensatie tussen de attributen optreedt, wat de extremen dempt. Een andere mogelijkheid is dat deze datasets gewoonweg geen grote verschillen laten zien, al lijkt het er op dat dit slechts een kleine kans heeft.

8 – Vergelijking met andere methoden

Het bekijken van klanttevredenheid aan de hand van de relatie tussen aspectbeoordelingen en een overall beoordeling, en eventuele verbeteringen aangeven met behulp van uninorms en de drie factor theorie is natuurlijk niet de enige manier om dergelijke analyses uit te voeren. In de literatuur zijn een aantal geaccepteerde methoden om het non-lineaire en asymmetrische verband tussen klanttevredenheid op een overall basis en op een attribuutbeoordelingen gestoelde basis te beoordelen, uninorms zijn pas recentelijk geïntroduceerd als mogelijkheid. Voorbeelden van alternatieve methoden zijn:

1. Critical Incident Technique (CIT)
2. Importance Performance Analysis (IPA)
3. Logaritmische regressie met dummyvariabelen (LRD)
4. Penalty-Reward Contrast Analysis (PRCA)

Deze alternatieve methoden zullen in het hier op volgende deel behandeld worden.

CIT stamt uit het psychologisch onderzoek en is gebaseerd op de positieve en/of negatieve feedback van gebruikers naar aanleiding van een gebeurtenis. Meestal wordt deze techniek toegepast wanneer een negatieve gebeurtenis plaats heeft gevonden zodat er kan worden gekeken wat er te verbeteren valt aan de respons na het voorval en wordt er gekeken naar de aanleiding van het voorval. “Critical Incident Technique” wordt vaak toegepast in een IT omgeving waar als gebeurtenis het falen van een applicatie wordt gedefinieerd, het is in deze hoedanigheid dan ook een zeer krachtige techniek om uiteindelijk de service te verbeteren. Ondanks de populariteit biedt deze techniek niet de mogelijkheid om de individuele invloed van een attribuut weer te geven, er wordt altijd gekeken naar de geaggregeerde invloed van alle attributen.

Een tweede manier is “Importance-Performance Analysis”, een techniek ontwikkeld in 1977 door Martilla en James (zie [7]). Hierbij wordt aan klanten via een vragenlijst een tweetal vragen gesteld over vooraf geselecteerde attributen van het product. Deze twee vragen hebben respectievelijk betrekking op de beoordeling van de mate van belangrijkheid van het attribuut voor de klant, en op de prestatie van het betreffende product op het attribuut in kwestie. Met deze informatie kan assenstelsel gemaakt worden waar de belangrijkheid en de prestatie uitgezet worden op de twee assen. Het ontstane viertal kwadranten kan gevuld worden met de geïnterpreteerde gegevens waarin de plaatsing van het punt natuurlijk gebaseerd is op de twee genoemde scores.

Deze methode is betrekkelijk eenvoudig, zowel in realisatie als in interpretatie, maar er zijn kanttekeningen bij geplaatst, onder andere door Matzler et al. (2004) [9]. Hier is een uitbreiding toegevoegd aan IPA waarbij er een tweede regressiecoëfficiënt in het leven is geroepen, de eerste meet de invloed wanneer de prestatie laag is, de tweede dat wanneer de prestatie goed is; zo is er onderscheid te maken tussen een tweetal soorten attributen.

Mede door de eenvoudige opzet gaat deze techniek uit van een asymmetrisch verband tussen prestaties en tevredenheid, iets wat niet wenselijk is. Beter zou het zijn geweest wanneer er geen enkele aanname over de vorm van de relatie gemaakt zou worden.

Bestudering van analyse met behulp van logaritmische regressie met dummy's (LRD) levert ons op dat dit een zeer effectieve manier is om aan de hand van een aantal variabelen aan te geven welke van hen het best de te voorspellen variabele verklaart. LRD is een techniek die inzicht kan verschaffen in hoeverre een waardering verklaard kan worden aan de hand van (on)afhankelijke variabelen, en zou een ranglijst kunnen opleveren van de deelaspecten gesorteerd op invloed ten aanzien van de te verklaren variabele.

Concreet: LRD kan een lijstje maken welke beoordeelde aspecten de meeste invloed op de eindwaardering hebben.

Mittal, et al. (zie [10]) hebben in 1998 een zeer uitgebreide studie gedaan naar het asymmetrische en non-lineaire verband tussen klanttevredenheid op basis van aspecten en op basis van het product als geheel. Men ging er van uit dat negatieve deelwaarderingen een groter effect hebben op het eindoordeel dan positieve deelwaarderingen. Zij kwamen tot de conclusie dat er wel degelijk een dergelijk verband is, en dat dit, zoals voorspeld, een asymmetrisch en non-lineair verband is. Een kanttekening die bij het onderzoek geplaatst kan worden is het feit dat er een non-lineair verband 'a priori' werd verondersteld, wat invloed op de resultaten kan hebben door het feit dat er alleen aan de mogelijkheid van een non-lineair model wordt gedacht.

Verder focust het onderzoek zich op het verband tussen klanttevredenheid op basis van aspecten en de overall-klanttevredenheid. Door deze opzet wordt er niet gekeken naar de invloed van een bepaald individu op de eindwaardering, maar puur en alleen naar de samenstelling van de eindwaardering op basis van de tussenbeoordelingen. Individuele voorkeuren en persoonlijke wegingen van aspecten worden niet meegenomen.

Deze laatste tekortkoming geeft LRD een grote achterstand op uninorm-analyse en kan daarom in een aantal gevallen als inferieur gezien worden.

Als laatste methode is "Penalty-Reward Contrast Analysis" (PRCA) een sterke methode die rekening houdt met individuele voorkeuren en attribuutwegingen van een individu. Matzler en Sauerwein (zie [8]) hebben empirisch aangetoond dat deze methode het beter doet dan de eerder behandelde IPA-methode. Ondanks dit is deze methode volgens Vanhoof [15] niet in staat alle mogelijke typen individuen (zie tabel 3.1) correct te behandelen

9 – Conclusie

Alhoewel de hier gebruikte datasets geen eenduidig antwoord wisten te produceren is het moeilijk om te twijfelen aan de potentiële kracht van analyse van klanttevredenheidsgegevens met behulp van uninorms en Kano's drie-factor theorie.

Een combinatie van deze twee geeft een krachtig gereedschap waarmee grafisch inzichtelijk kan worden gemaakt welke aspecten het meest van belang zijn voor een klant. Wanneer een manager er voor zorgt dat de belangrijkste aspecten een optimale waarde krijgen, is er een behoorlijke winst behaald tegenover een maximalisatie van de aspecten. Dit laatste is in de meeste gevallen duurder en minder kosteneffectief, gezien het feit dat klanttevredenheid vaak te kampen heeft met afnemende meeropbrengsten

Natuurlijk is het niet alleen goud wat er blinkt, ook de hier besproken theorie heeft tekortkomingen.

Zo houdt de huidige opzet van het onderzoek geen rekening met eventuele weging van 'belangrijkheid' van een attribuut. Momenteel zijn alle aspecten voor de klant even belangrijk in de theorie, terwijl dit niet zo hoeft te zijn, en in de praktijk bijna nooit zo is. Deze weging kan geïntroduceerd worden door de ondervraagde tegelijkertijd met het geven van de beoordelingen een extra beoordeling van de belangrijkheid van het attribuut op een schaal van bijvoorbeeld 1 tot 5 te laten geven. Belangrijk is het hier dat deze schaal niet te groot wordt, aangezien er dan incorrecte en onwenselijke antwoorden kunnen ontstaan doordat de ondervraagde het overzicht kwijtraakt, of (erger nog) de interesse verliest. Zie hiervoor Babbie [1]. Deze extra weging zorgt er voor dat de gegenereerde uninorm beter aansluit op de te beoordelen data en daardoor een correctere schatting kan geven van de neutrale waarde van een klant.

Ter verduidelijking: formule (4) zou aangepast moeten worden tot

$$U(s_1 * w_1, \dots, s_n * w_n) = S, \quad (13)$$

waarbij $[w_1, \dots, w_n]$ de waarderingen van belangrijkheid weergegeven van het attribuut voor de klant. Deze wegingen liggen in theorie tussen 0 en ∞ , alhoewel het in de praktijk waarschijnlijk op het interval $[0.1, 2]$ zal liggen.

Een ander punt voor verbetering is het feit dat momenteel de neutrale waarde voor een klant als geheel is, en niet per attribuut. Het is zeer wel mogelijk dat een klant een totaal andere verwachting heeft ten aanzien van attribuut A dan van attribuut B. In de huidige opzet is dit van weinig invloed op de resultaten, en wordt het slechts beperkt weergegeven in de uninorm door de uitwerking die het heeft op de neutrale waarde van de klant als geheel.

Een oplossing die gegeven kan worden is tweeledig, de eerste oplossing vereist een ingrijpende verandering van het onderzoek. Wanneer er historische data beschikbaar is van de klant in kwestie van een productevaluatie hetzelfde product en/of dienst of een evaluatie van een vergelijkbaar aspect kan deze

gebruikt worden om een historisch gemiddelde af te lezen wat een indicatie geeft van de voorkeuren en eisen van de klant op dat specifieke aspect. Met deze informatie kan de uninorm ook beter aangepast worden zodat deze zo optimaal mogelijk aansluit bij de verstrekte data.

Een tweede oplossing zou het bundelen van verscheidene vergelijkbare aspecten zijn tot een 'beoordelingsgebied'. Individuele beoordelingen kunnen dan geaggregeerd worden tot een voorkeur die per beoordelingsgebied berekend en toegepast kan worden. Ook hiermee wordt gepoogd de uninorm beter aan te laten sluiten op de persoonlijke voorkeuren van een ondervraagde.

Eventueel kunnen beide hierboven gepresenteerde oplossingen gecombineerd worden om een maximaal resultaat te behalen.

Vervolgonderzoeksmogelijkheden zijn er legio, waarbij er gedacht kan worden aan toepassing van bovenstaande verbeteringen van de analyse. Ook zijn er, zeker in dit specifieke onderzoek, nog technieken toepasbaar waarmee de kwaliteit en de kwantiteit van de data te verbeteren is. Zo bestaat de mogelijkheid om de onvermijdelijke onvolledige beoordelingen niet gewoon buiten beschouwing te laten zoals hier is gedaan, maar om de ontbrekende waarden aan te passen tot een (gewogen) gemiddelde van de overige respondenten. Ook is het mogelijk om de dataset te vergroten door te kijken of het weglaten van een aspect een grotere dataset oplevert, met minder ontbrekende waarden.

Deze laatste twee dataverrijkingstechnieken zijn zeer ingrijpend en leiden mogelijk tot een totaal veranderde dataset. Zeker bij de laatste beschreven techniek bestaat de kans dat er een vitaal onderdeel van de dataset buiten beschouwing wordt gelaten, iets wat absoluut onwenselijk is.

Verder is er vervolgonderzoek gewenst op basis van minder brede datasets met meer beoordelingen dan de data die in dit onderzoek zijn gebruikt. Waarschijnlijk worden de effecten extremer en zal er minder compensatie tussen de aspecten onderling voorkomen. Deze compensatie bemoeilijkte in dit onderzoek de interpretatie van de gegevens.

10 – Literatuurlijst

- [1] E. Babbie (1975). *The practice of Social Research*, Thomson Wadsworth. Eleventh International Student Edition, 2006.
- [2] G. H. van Bruggen, A. Smidts, B. Wierenga, (1998). Improving Decision Making by Means of a Marketing Decision Support System, *Management Science*, Volume 44, Issue 5, pp. 645-658.
- [3] T. Calvo, R. Mesiar, (2003). Weighted triangular norms-based aggregation operators, *Fuzzy Sets and Systems*, Volume 137, Issue 1, pp. 3-10.
- [4] T. Calvo, R. Mesiar, (2003). Aggregation operators: ordering and bounds, *Fuzzy Sets and Systems*, Volume 139, Issue 3, pp. 685-697.
- [5] J. Dombi, (1982). Basic concepts for a theory of evaluation: The aggregative operator, *European Journal of Operational Research*, Volume 10, pp. 282-293.
- [6] D. Dubois, H. Prade, (1985). A review of fuzzy set aggregation connectives, *Information Sciences*, Volume 35, Issues 1-2, pp. 85-121.
- [7] J. A. Martilla, J. C. James (1977). Importance-Performance Analysis, *Journal of Marketing*, Volume 41, Issue 1, pp. 77-79.
- [8] K. Matzler, E. Sauerwein (2002). The factor structure of customer satisfaction: An empirical test of the importance grid and the penalty-reward-contrast analysis, *International Journal of Service Industry Management*, Volume 13, Issue 4, pp. 314-332.
- [9] K. Matzler, F. Bailom, H. H. Hinterhuber, B. Renzl, J. Pichler, (2004). The asymmetric relationship between attribute-level performance and overall customer satisfaction: a reconsideration of the importance–performance analysis, *Industrial Marketing Management*, Volume 33, Issue 4, pp. 271-277.
- [10] V. Mittal, W.T. Ross Jr., P.M. Baldasare, (1998). The Asymmetric Impact of Negative and Positive Attribute-Level Performance on Overall Satisfaction and Repurchase Intentions, *Journal of Marketing*, Volume 62, pp. 33-47.
- [11] J. H. Myers, M. I. Alpert (1968). Determinant Buying Attitudes: Meaning and Measurement, *Journal of Marketing*, Volume 32, Issue 4, pp. 13-20.
- [12] Y. Li and Z. Shi, (2000). Remarks on uninorm aggregation operators, *Fuzzy Sets and Systems*, Volume 114, Issue 3, pp. 377-380.
- [13] J. E. Swan, L. J. Combs (1976). Product Performance and Consumer Satisfaction: A New Concept, *Journal of Marketing*, Volume 40, Issue 2, pp 25-32.
- [14] A. Tversky, D. Kahneman (1974). Judgement under Uncertainty: Heuristics and Biases, *Science*, Volume 185, pp. 1124-1131.

- [15] K. Vanhoof, P. Pauwels, J. Dombi, T. Brijs, G. Wets, (2005). Penalty-Reward Analysis with Uninorms: A Study of Customer (Dis)Satisfaction, *Studies in Computational Intelligence*, Volume 5, pp. 237-252.
- [16] R. R. Yager (1994). Responses to Elkan (Ronald R. Yager), *IEEE Expert: Intelligent Systems and Their Applications*, Volume 9, Issue 4, pp. 41-42.
- [17] R. R. Yager, A. Rybalov (1996). Uninorm aggregation operators, *Fuzzy Sets and Systems*, Volume 80, Issue 1, pp. 111-120.
- [18] R. R. Yager, A. Rybalov (1998). Full Reinforcement Operators in Aggregation Techniques, *IEEE Transactions on Systems, Man and Cybernetics*, Volume 28, pp. 757-769.

11 – Bijlagen

11.1 – Implementatie

Voor het onderzoek is er een programma geschreven om de berekeningen met betrekking tot de uninorms volledig geautomatiseerd uit te voeren. Dit is gedaan in de programmeertaal Java, mede ingegeven door het feit dat ik zelf enige ervaring heb met het werken met die taal door mijn vooropleiding.

Het programma is in essentie opgebouwd uit slechts 1 “package”, UniNorms genoemd. Hierin zitten een vijftal klassen, te weten UniNorm, Process, PreProcessor, Response en Calculator. Ik heb gepoogd alles zo modulair mogelijk te houden, en te werken volgens de principes van Object Georiënteerd Programmeren. (OOP, Object Oriented Programming)

Het tweede package wat is toegevoegd zijn hulpmiddelen, te weten het package Utils. Dit bestaat uit slechts 1 klasse: Utils.RandomCreator.

Wanneer we in vogelvlucht het programma zouden bekijken zou de klasse Uninorms slechts een goed begin zijn van het hele proces van berekeningen. De belangrijkste variabelen worden ingelezen vanuit een bestand waarna alles door de klasse Process wordt verwerkt. Hier wordt de data ingelezen vanaf de harde schijf en via de PreProcessor met behulp van de in hoofdstuk 5 beschreven deling op het juiste interval gezet. Direct na het inlezen wordt er per ingelezen regel een Response aangemaakt waar de data in gaat. Als alle “responses” zijn gecreëerd worden ze stuk voor stuk aangeboden aan de “Calculator” die alle berekeningen uitvoert. Hier worden de neutrale waarde, de alpha en uiteindelijk de individuele bijdragen berekend, alles volgens de in hoofdstuk 4 beschreven methoden.

Terugkomend op de wiskundige achtergrond, er is gekozen voor een generatorfunctie $g(x) = \ln\left(\frac{x}{1-x}\right)$,

en de theorie is uitgeschreven tot $\alpha = \frac{1}{n-1} \left(\sum_{i=1}^n g(s_i) - g(S) \right)$.

Wanneer we α willen berekenen moet de formule die we hebben gekozen als generatorfunctie ingevuld

worden in (5), wat $\alpha = \frac{1}{n-1} \left(\sum_{i=1}^n \ln \frac{s_i}{1-s_i} - \ln \frac{S}{1-S} \right)$ oplevert.

Ter herhaling: $g(x) = f^{-1}(x)$, en dus $f(x) = g^{-1}(x)$.

Toegepast op onze huidige $f(x)$ geeft dit $v = \frac{e^\alpha}{1+e^\alpha}$, aangezien $f(x) = \frac{e^x}{1+e^x}$.

Met de nu bekende α kan ν berekend worden volgens (7). Dit levert ons $\nu = \frac{e^\alpha}{1+e^\alpha}$ op.

De volgende stap is het berekenen van de bijdrage. De twee termen in $\text{Contrib}(j,i)$, (8), behandelen we apart. We beginnen met het eerste deel, te weten $U(s_1, \dots, s_{j-1}, s_j, s_{j+1}, \dots, s_n)$, welke gelijk is aan $g_\alpha^{-1}\left(\sum_{i=1}^n g_\alpha(s_i)\right)$. Deze laatste term is al een bekende, dit is namelijk per definitie gelijk aan de uninorm, en daarmee dus gelijk aan de eindwaardering S .

De volgende term in de bijdrage is lastiger, te weten $U(s_1, \dots, s_{j-1}, \nu_i, s_{j+1}, \dots, s_n)$.

Dit is gelijk aan $g^{-1}\left(\sum_{k=1}^j g_\alpha(s_k) + g_\alpha(\nu_i) + \sum_{k=j+1}^n g_\alpha(s_k)\right)$.

Hieruit volgt $g^{-1}\left(\sum_{k=1}^j g_\alpha(s_k) + g_\alpha(\nu_i) + \sum_{k=j+1}^n g_\alpha(s_k)\right) = g^{-1}\left(\sum_{k=1}^n g_\alpha(s_k) - g_\alpha(s_j) + g_\alpha(\nu_i) + \alpha\right)$

waarbij we $\sum_{k=1}^j g_\alpha(s_k) + \sum_{k=j+1}^n g_\alpha(s_k)$ hebben vervangen door $\sum_{k=1}^n g_\alpha(s_k) - g_\alpha(s_j)$ aangezien dit equivalenten zijn.

De volgende stap is het omschrijven van $g_\alpha(x)$ naar $g(x) - \alpha$. Ook wordt $g_\alpha(\nu_i)$ vervangen door 0

daar deze per definitie gelijk aan elkaar zijn. Dit levert ons $g^{-1}\left(\sum_{k=1}^n g_\alpha(s_k) - g_\alpha(s_j) + g_\alpha(\nu_i) + \alpha\right) =$
 $= g^{-1}\left(\left(\sum_{k=1}^n g(s_k) - n\alpha\right) - (g(s_j) - \alpha) + 0 + \alpha\right)$ op.

Na het samenvoegen van de α 's komen we uit op $g^{-1}\left(\sum_{k=1}^n g(s_k) - g(s_j) + (2-n)\alpha\right)$.

Gegeven het feit dat $(n-1)\alpha = \sum_{i=1}^n g(s_i) - g(S)$ en $g^{-1}(x + \alpha) = g_\alpha^{-1}(x)$ komen we via

$$\begin{aligned} & g^{-1}\left(\sum_{k=1}^n g(s_k) - g(s_j) + (2-n)\alpha\right) = \\ & = g^{-1}\left((g(S) + (n-1)\alpha) - g(s_j) + (2-n)\alpha\right) = \\ & = g^{-1}\left(g(S) - g(s_j) + \alpha\right) = \\ & = g_\alpha^{-1}\left(g(S) - g(s_j)\right) \end{aligned}$$

tot

$$g_\alpha^{-1} \left(\ln \left(\frac{S}{1-S} \right) - \ln \left(\frac{s_j}{1-s_j} \right) \right). \quad (14)$$

Wetende dat $f(x) = \frac{e^x}{1+e^x} = \frac{1}{1+e^{-x}}$, en dat natuurlijk $f_\alpha(x) = f(x+\alpha)$ ook nog steeds geldt, (wat

$f_\alpha(x) = \frac{1}{1+e^{-(x+\alpha)}}$ oplevert), kan (14) omgeschreven worden tot

$$\frac{1}{1+e^{-\left(\ln\left(\frac{S}{1-S}\right) - \ln\left(\frac{s_j}{1-s_j}\right) + \alpha\right)}}. \quad (15)$$

De formule voor de bijdrage is dan ook na combinatie van (4) en (15) als volgt:

$$\text{Contrib}(j,i) = S - \frac{1}{1+e^{-\left(\ln\left(\frac{S}{1-S}\right) - \ln\left(\frac{s_j}{1-s_j}\right) + \alpha\right)}} \quad (16)$$

Voor alle duidelijkheid volgt hierna een overzicht met uitwerkingen van de meest voorkomende formules, dit om misverstanden met inverse formules en “displaced” formules te voorkomen.

f	g
$f(x) = \frac{e^x}{1+e^x}$	$g(x) = \ln\left(\frac{x}{1-x}\right)$
$f^{-1}(x) = \ln\left(\frac{x}{1-x}\right)$	$g^{-1}(x) = \frac{e^x}{1+e^x}$
$f_\alpha(x) = \frac{e^{x+\alpha}}{1+e^{x+\alpha}}$	$g_\alpha(x) = \ln\left(\frac{x}{1-x}\right) - \alpha$
$f_\alpha^{-1}(x) = \ln\left(\frac{x}{1-x}\right) - \alpha$	$g_\alpha^{-1}(x) = \frac{e^{x+\alpha}}{1+e^{x+\alpha}}$

Tabel 11.1.1

11.2 – Programmacode

11.2.1 – UniNorms.UniNorm

```
package UniNorms;
import java.io.BufferedReader;
import java.io.FileReader;
/**
 * The UniNorm class is the main class in this package.
 * It initiates the calculation of the neutral values of the responses in the
 * designated file.
 * <p>
 * The parameters are hard-coded in the source.
 * The standards are:
 * <ul>
 * <li>FILENAME: data.csv</li>
 * <li>MAX: [1:10]</li>
 * <li>NUMBEROFATTRIBUTES: 3</li>
 * <li>WRITEFILE: output/data_out.csv</li>
 * <li>SETTINGSFILENAME: UniNorms.ini</li>
 * </ul>
 * These can ofcourse be modified if needed.
 * The values in SETTINGSFILENAME will be used if present.
 *
 * @author Maarten Tijhof (275772mt@eur.nl)
 * @version 0.4 (2006-03-15)
 */
class UniNorm {

    private String FILENAME;
    private double MAX;
    private double NUMBEROFATTRIBUTES;
    private static final String SETTINGSFILENAME = "UniNorms.ini";
    private Process process;
    private String WRITEFILE;

    /**
     * The constructor for the class UniNorm.
     * It first sets some values to their standard values, and the tries to read
     * a file containing overriding values. If it is not found, it outputs the
     * info to the screen and continues with the default values.
     * When the file is found, the values are used in the process.
     */
    public UniNorm() {
        FILENAME = "data.csv";
        MAX = 10;
        NUMBEROFATTRIBUTES = 3;
        WRITEFILE = "output\\out.csv";
        try {
            BufferedReader in = new BufferedReader(new FileReader(SETTINGSFILENAME));
            while(in.ready()) {
                String line = in.readLine();
                if(!line.substring(0,0).equals(";")) {
                    String[] read = line.split("=");
                    if(read[0].trim().equals("FILENAME")) {
                        FILENAME = read[1].trim();
                    } else if(read[0].trim().equals("MAX")) {
                        MAX = new Double(read[1].trim()).doubleValue();
                    } else if(read[0].trim().equals("NUMBEROFATTRIBUTES")) {
                        NUMBEROFATTRIBUTES = new Double(read[1].trim()).doubleValue();
                    } else if(read[0].trim().equals("WRITEFILE")) {
                        WRITEFILE = read[1].trim();
                    }
                }
            }
        }
        catch(Exception e) {
            System.out.println("Something went horribly wrong whilst reading the file " + SETTINGSFILENAME);
        }
    }
}
```

```
        System.out.println(" The returned error is: " + e);
        System.out.println("Now trying to run the program with the default settings...");
    }
    process = new Process(FILENAME, NUMBEROFATTRIBUTES, MAX);
    process.printToFile(WRITEFILE);
    process.printAggregateInfoToFile(WRITEFILE);
}

/**
 * The main-method.
 * Creates a new instance of Process and initiates it using the hard-coded
 * values for FILENAME, MAX and NUMBEROFATTRIBUTES.
 */
public static final void main(String[] args) {
    UniNorm uniNorm = new UniNorm();
}
}
```

11.2.2 – UniNorms.Process

```
package UniNorms;
import java.io.BufferedReader;
import java.io.FileReader;
import java.util.ArrayList;
import java.util.Iterator;
import java.io.PrintWriter;
import java.io.BufferedWriter;
import java.io.FileWriter;
/**
 * Process reads the input-file and processes all responses accordingly.
 *
 * @author Maarten Tijhof (275772mt@eur.nl)
 * @version 0.8 (2006-03-20)
 */
public class Process {

    private ArrayList responses;
    private ArrayList contents;
    private double MAX;
    private double NUMBEROFATTRIBUTES;
    private Calculator calc;
    private double[][] aggregateInfo;

    /**
     * Constructor for objects of class Process.
     * It initialises a filereader which is to process the input through a calculator.
     * <p>
     * Some restrictions apply to the datafile:
     * <ul>
     * <li>It must be CSV with a semicolon ";" as delimiter</li>
     * <li>Each line has to have at least 2 values</li>
     * <li>The last value of a line is the overall satisfactionlevel</li>
     * <li>The data must be complete and uncorrupted</li>
     * <li>All values must be on the interval [1:MAX] (Inclusive)</li>
     * </ul>
     *
     * @param FILENAME      The relative path to the datafile.
     * @param MAX           The maximum of the interval the grades are on.
     *                     [1:MAX]. (Inclusive)
     * @param NUMBEROFATTRIBUTES The number of attributes (excluding the overall satisfaction) of
     *                     each response.
     */
    public Process(String FILENAME, double NUMBEROFATTRIBUTES, double MAX) {
        this.MAX = MAX;
        this.NUMBEROFATTRIBUTES = NUMBEROFATTRIBUTES;
        contents = new ArrayList();
        responses = new ArrayList();
        calc = new Calculator();
        PreProcessor preProcessor = new PreProcessor(FILENAME, NUMBEROFATTRIBUTES, MAX);
        FILENAME = preProcessor.start();
    }
}
```

```
    try {
        BufferedReader in = new BufferedReader(new FileReader(FILENAME));
        while(in.ready()) {
            contents.add(in.readLine());
        }
    } catch(Exception e) {
        System.out.println("Something went horribly wrong whilst reading the file " + FILENAME);
        System.out.println(" The returned error is: " + e);
    }
}
process();
}

/**
 * This method processes the read lines from the file FILENAME.
 * It processes the input through a calculator.
 * The results are being printed to the screen, not (yet) written to a file.
 */
public void process() {
    Iterator it = contents.iterator();
    int count = 0;
    boolean onlyLastOne = false;
    double beta = 0.5;
    while(it.hasNext()) {
        count++;
        String entry = (String)it.next();
        String[] tmp = entry.split(";");
        double[] values = new double[tmp.length-1];
        for(int i=0; i<(tmp.length-1); i++) {
            double temp = 0;
            if(tmp[i].length() != 0) {
                temp = new Double(tmp[i]).doubleValue();
                if(temp == MAX && onlyLastOne) {
                    values[i] = temp/(MAX+beta);
                } else {
                    values[i] = temp/MAX;
                }
            } else {
                values[i] = 0;
            }
        }
        double endTemp = 0;
        if(tmp[tmp.length-1].length() != 0) {
            endTemp = new Double(tmp[tmp.length-1]).doubleValue();
        }
        double endValue = 0;
        if(onlyLastOne && endTemp == MAX) {
            endValue = endTemp/(MAX+beta);
        } else {
            endValue = (endTemp/MAX);
        }
        Response resp = new Response(count,values,endValue);
        responses.add(resp);
        calc.calculate(resp);
//        resp.printInfoVHS(MAX);
    }
    sortProcessedResponses();
}

/**
 * This method outputs the info of each response to a CSV-file.
 * This file also contains the original info, all in the well known format.
 *
 * @see Response#getPrintInfo
 * @param WRITEFILE The file this method will write to.
 */
public void printToFile(String WRITEFILE) {
    PrintWriter out = null;
    try {
        out = new PrintWriter(new BufferedWriter(new FileWriter(WRITEFILE)), true);
    } catch(Exception e) {
        System.out.println("Something went horribly wrong whilst initiating writing the file " + WRITEFILE);
    }
}
```

```

        System.out.println(" The returned error is: " + e);
    }
    try {
        Response response = (Response)responses.get(0);
        out.print("Number;");
        for(int i=1; i<=response.getValues().length; i++) {
            out.print("value["+i+"];contrib["+i+"];");
        }
        out.println("endValue;Neutral");
        Iterator it = responses.iterator();
        while(it.hasNext()) {
            response = (Response)it.next();
            //out.println(response.getPrintInfoVHS(MAX));
            out.println(response.getPrintInfo());
        }
    } catch(Exception e) {
        System.out.println("Something went horribly wrong whilst writing to the file " + WRITEFILE);
        System.out.println(" The returned error is: " + e);
    }
}

/**
 * This method calculates on a per attribute-value basis the average of the contribution.
 * So, for every possible value of an attribute this method calculates the average contribution
 * of that value towards the overall-satisfaction.
 * <p>
 * These values are stored in a multi-dimensional array, and are fully prepped for output.
 * Additionally, an error-check has been built in to make sure all values are processed, and none are left out.
 * As a side-effect, it also re-ensures us of the fact all data is complete and uncorrupted.
 * When an anomaly is found, the process is stopped. Further action must then be taken towards either the program or the
data.
 * </p>
 */
public void sortProcessedResponses() {
    Response temp = (Response)responses.get(0);
    double[][] array = new double[temp.getValues().length][new Double(MAX).intValue()];
    int[][] freq = new int[temp.getValues().length][new Double(MAX).intValue()];
    Iterator it = responses.iterator();
    int i = 0;
    while(it.hasNext()) {
        Response resp = (Response)it.next();
        double[] values = resp.getValues();
        for(int j=0; j<values.length; j++) {
            int value = (new Double(Math rint(values[j]*MAX)).intValue())-1;
            array[j][value] += resp.getContribution((j+1));
            freq[j][value]++;
        }
        i++;
    }
    int sum = 0;
    for(int j=0; j<array.length; j++) {
        for(int k=0; k<array[j].length; k++) {
            if(freq[j][k] == 0) {
                array[j][k] = 0;
            } else {
                array[j][k] = array[j][k]/freq[j][k];
            }
            sum += freq[j][k];
            System.out.print(array[j][k] + "\t");
        }
        System.out.println();
    }
    if(sum != (i*NUMBEROFATTRIBUTES)) {
        System.out.println("The data has become corrupted in the process.");
        System.out.println("Further processing will not be done, the program will end.");
        System.exit(0);
    }
    aggregateInfo = array;
}

/**

```

```
* This method outputs the aggregate info of all contributions to a file.
*
* @param WRITEFILE    The file this method will write to, in a slightly formatted way.
*/
public void printAggregateInfoToFile(String WRITEFILE) {
    String[] splitName = WRITEFILE.split("\\.");
    WRITEFILE = splitName[0].concat("_aggr.").concat(splitName[1]);
    PrintWriter out = null;
    try {
        out = new PrintWriter(new BufferedWriter(new FileWriter(WRITEFILE)), true);
    } catch(Exception e) {
        System.out.println("Something went horribly wrong whilst initiating writing the file " + WRITEFILE);
        System.out.println(" The returned error is: " + e);
    }
    try {
        out.print("Number");
        for(int i=1; i<=aggregateInfo[0].length; i++) {
            out.print(";value["+i+"]");
        }
        out.println();
        for(int i=0; i<aggregateInfo.length; i++) {
            out.print("("+(i+1));
            for(int j=0; j<aggregateInfo[i].length; j++) {
                out.print("; " + (new Double(aggregateInfo[i][j])).toString().replaceAll("\\.", ","));
            }
            out.println();
        }
    } catch(Exception e) {
        System.out.println("Something went horribly wrong whilst writing to the file " + WRITEFILE);
        System.out.println(" The returned error is: " + e);
    }
}
}
```

11.2.3 – UniNorms.PreProcessor

```
package UniNorms;
import java.io.BufferedReader;
import java.io.FileReader;
import java.io.PrintWriter;
import java.io.BufferedWriter;
import java.io.FileWriter;
import java.util.ArrayList;
import java.util.Iterator;
/**
 * The PreProcessor goes through the data and tries to edit the data
 * to match the prerequisites set by the class {@link UniNorm}.
 * <p>
 * These include:
 * <ul>
 * <li>The data must be complete, e.g. no values can be missing</li>
 * <li>The data must be uncorrupted, e.g. no non-numbers may be present</li>
 * </ul>
 * </p>
 * As of version 0.2 (2006-03-16) the preprocessor just deletes all incorrect lines and goes
 * on processing all others, written to a temporary file.
 * This file is named &lt;originalName&gt;_temp.&lt;originalExt&gt;, and can be found in the base directory..
 *
 * @author Maarten Tijhof (275772mt@eur.nl)
 * @version 0.2 (2006-03-16)
 */
public class PreProcessor
{
    private String FILENAME;
    private double MAX;
    private double NUMBEROFATTRIBUTES;
    private ArrayList contents;
    private Double[] average;
    private ArrayList newContents;
```

```
/**
 * Constructor for objects of class PreProcessor.
 *
 * All needed values are being initiated here, and will be ready for use.
 * Does nothing more really.
 *
 * @param FILENAME      The relative path to the datafile.
 * @param MAX           The maximum of the interval the grades are on.
 *                     [1:MAX]. (Inclusive)
 * @param NUMBEROFATTRIBUTES The number of attributes (excluding the overall satisfaction) of
 *                     each response.
 */
public PreProcessor(String FILENAME, double NUMBEROFATTRIBUTES, double MAX) {
    this.FILENAME = FILENAME;
    this.MAX = MAX;
    this.NUMBEROFATTRIBUTES = NUMBEROFATTRIBUTES;
    contents = new ArrayList();
}

/**
 * This method actually does all the hard work necessary to get all inconsistencies and missing values out of the data.
 *
 * It checks for missing values on all places, whether they are a satisfactionlevel of an attribute, or an overall satisfaction.
 * If such an inconsistency is found, this method will ignore this line in whole, in effect deleting it from the data.
 * When all data is examined, and at least one inconsistency is found, it will write all syntactically correct data
 * to a temporary file.
 * Special occasion is the detection of a non-numeric value, this is printed to the screen. The line will still be ignored.
 * The (modified) filename FILENAME is then returned to the method calling start().
 *
 * @return String The name of the file the (modified) data is located in.
 */
public String start() {
    int count = 0;
    try {
        BufferedReader in = new BufferedReader(new FileReader(FILENAME));
        Double test = null;
        while(in.ready()) {
            boolean foundEmptyValues = false;
            String[] line = in.readLine().split(";");
            if(line.length <= NUMBEROFATTRIBUTES) {
                foundEmptyValues = true;
            } else {
                for(int i=0; i<line.length; i++) {
                    if(line[i].equals("")) {
                        foundEmptyValues = true;
                        i = line.length;
                    } else {
                        try {
                            test = new Double(line[i]);
                        } catch(NumberFormatException e){
                            System.out.println("There seems to be a problem with your data.");
                            System.out.println("I think there might be non-numeric values within the data itself.");
                            System.out.println("The one I found lives at row "+(count+1)+" , and column "+(i+1)+" .");
                            System.out.println("Please correct this.");
                            System.out.println(" The returned error is: " + e);
                            //System.exit(0);
                            foundEmptyValues = true;
                            i = line.length;
                        }
                    }
                }
            }
            if(!foundEmptyValues) {
                contents.add(line);
            }
            count++;
        }
    } catch(Exception e) {
        System.out.println("Something went horribly wrong whilst reading the file " + FILENAME);
        System.out.println(" The returned error is: " + e);
    }
}
```

```
}
if(contents.size() != count) {
    String[] splitName = FILENAME.split("\\.");
    String FILENAMETEMP = splitName[0].concat("_temp.").concat(splitName[1]);
    PrintWriter out = null;
    try {
        out = new PrintWriter(new BufferedWriter(new FileWriter(FILENAMETEMP)), true);
    } catch(Exception e) {
        System.out.println("Something went horribly wrong whilst initiating writing the file " + FILENAMETEMP);
        System.out.println(" The returned error is: " + e);
    }
    try {
        Iterator it = contents.iterator();
        while(it.hasNext()) {
            String[] toPrint = (String[])it.next();
            for(int i=0; i<toPrint.length; i++) {
                out.print(toPrint[i]+";");
            }
            out.println();
        }
    } catch(Exception e) {
        System.out.println("Something went horribly wrong whilst writing to the file " + FILENAMETEMP);
        System.out.println(" The returned error is: " + e);
    }
    FILENAME = FILENAMETEMP;
}
return FILENAME;
}

/**
 * This method is mostly equal to start().
 * The only difference is that it will stop when a non-numeric value is found,
 * and that it will replace all missing values with the average value of the attribute.
 *
 * It is much more memory-expensive, and will not be used (anymore)
 * Consider it legacy.
 * (as of 2006-03-15)
 */
private String modifyData() {
    try {
        BufferedReader in = new BufferedReader(new FileReader(FILENAME));
        while(in.ready()) {
            String[] line = in.readLine().split(";");
            contents.add(line);
        }
    } catch(Exception e) {
        System.out.println("Something went horribly wrong whilst reading the file " + FILENAME);
        System.out.println(" The returned error is: " + e);
    }
    int attNum = ((String[])contents.get(0)).length;
    average = new Double[attNum];
    Iterator it = null;
    for(int i=0; i<attNum; i++) {
        it = contents.iterator();
        double sum = 0;
        int count = 0;
        while(it.hasNext()) {
            try {
                String numberToBe = ((String[])it.next())[i];
                //if(numberToBe != "") {
                    sum += new Double(numberToBe);
                    count++;
                //}
            } catch(NumberFormatException e){
                //System.out.println("There seems to be a problem with your data.");
                //System.out.println("I think there might be non-numeric values within the data itself.");
                //System.out.println("The one I found lives at row "+(count+1)+", and column "+(i+1)+".");
                //System.out.println("Please correct this.");
                //System.out.println(" The returned error is: " + e);
                //System.exit(0);
            }
        }
    }
}
```



```
    }
    average[i] = (sum/count);
  }
  it = contents.iterator();
  newContents = new ArrayList();
  boolean foundEmptyValues = false;
  while(it.hasNext()) {
    String[] array = (String[])it.next();
    Double[] doubles = new Double[attNum];
    for(int i=0; i<attNum; i++) {
      if(array[i].equals("")) {
        doubles[i] = new Double(average[i]);
        foundEmptyValues = true;
      } else {
        doubles[i] = new Double(array[i]);
      }
    }
    newContents.add(doubles);
  }
  if(foundEmptyValues) {
    String[] splitName = FILENAME.split("\\.");
    String FILENAMETEMP = splitName[0].concat("_temp.").concat(splitName[1]);
    PrintWriter out = null;
    try {
      out = new PrintWriter(new BufferedWriter(new FileWriter(FILENAMETEMP)), true);
    } catch(Exception e) {
      System.out.println("Something went horribly wrong whilst initiating writing the file " + FILENAMETEMP);
      System.out.println(" The returned error is: " + e);
    }
    try {
      it = newContents.iterator();
      while(it.hasNext()) {
        Double[] toPrint = (Double[])it.next();
        for(int i=0; i<toPrint.length; i++) {
          out.print(toPrint[i]+";");
        }
        out.println();
      }
    } catch(Exception e) {
      System.out.println("Something went horribly wrong whilst writing to the file " + FILENAMETEMP);
      System.out.println(" The returned error is: " + e);
    }
    FILENAME = FILENAMETEMP;
  }
  return FILENAME;
}
}
```

11.2.4 – UniNorms.Response

```
package UniNorms;
/**
 * A class representing the responses of one of the questionees.
 * Actually just a wrapper-class to make life easier programming-wise.
 * <p>
 * Some restrictions do apply on the data itself:
 * <ul>
 * <li>"values" has to be a {@link double}-array with at least one value in it</li>
 * <li>"endValue" has to be a {@link double}</li>
 * <li>Both the double-array and the double must be on the interval [0:1] (inclusive)</li>
 * </ul>
 * <p>
 * After processing it also holds the calculated values like the neutral value
 * and all contributions.
 *
 * @author Maarten Tijhof (275772mt@eur.nl)
 * @version 0.5 (2006-03-20)
 */
public class Response
```

```
{
    private int number;
    private double[] values;
    private double endValue;
    private double[] contributions;
    private double neutral;

    /**
     * Constructs a response with the supplied values in it.
     *
     * @param number The assigned number of this response.
     * @param values A double-array with all satisfaction-levels in it.
     * @param endValue The given mark for the product as a whole.
     */
    public Response(int number, double[] values, double endValue) {
        this.number = number;
        this.values = values;
        this.endValue = endValue;
    }

    /**
     * Returns the number of this particular response.
     *
     * @return An int stating the number of this response.
     */
    public int getNumber() {
        return number;
    }

    /**
     * Returns the double-array with all satisfactionlevels.
     *
     * @return A double-array with all satisfactionlevels.
     */
    public double[] getValues() {
        return values;
    }

    /**
     * Returns the overall satisfactionlevel.
     *
     * @return The double with the overall satisfactionlevel.
     */
    public double getEndValue() {
        return endValue;
    }

    /**
     * Sets the contributions to the endValue based on the satisfactionlevels
     * per attribute.
     *
     * @param contributions A double-array with all contributions. Must be equal in size to
     * values in {@link #getValues()}.
     */
    public void setContributions(double[] contributions) {
        this.contributions = contributions;
    }

    /**
     * Sets the neutral value. Nothing more to it really.
     *
     * @param neutral The actual calculated neutral value.
     */
    public void setNeutral(double neutral) {
        this.neutral = neutral;
    }

    /**
     * Prints the information of this Response to screen.
     * <p>
     * It is in the following format:

```

```
* <code>&lt;value[i]&gt; &lt;contribution[i]&gt; ... &lt;value[n]&gt; &lt;contribution[n]&gt; &lt;endValue&gt;
&lt;neutral&gt;</code></p>
*/
public void printInfo() {
    for(int i=0; i<values.length; i++) {
        System.out.print(values[i] + "\t" + contributions[i] + "\t");
    }
    System.out.println(endValue + "\t" + neutral);
}

/**
 * Prints the information of this Response to screen, according to the format Vanhooff specified.
 * <p>
 * It is in the following format:
 * <code>&lt;value[i]&gt; &lt;contribution[i]&gt; ... &lt;value[n]&gt; &lt;contribution[n]&gt; &lt;endValue&gt;
&lt;neutral&gt;</code></p>
 *
 * @param MAX The maximum value of an individual attribute.
 */
public void printInfoVHS(double MAX) {
    for(int i=0; i<values.length; i++) {
        System.out.print((values[i]*MAX) + "\t" + Math rint(contributions[i]*MAX*10)/10 + "\t");
    }
    System.out.println((endValue*MAX) + "\t" + Math rint(neutral*MAX*10)/10);
}

/**
 * This method returns only 1 specific Contributionvalue specified by the
 * parameter number.
 *
 * @param number An integer on the interval [1:max] (inclusive) where max
 * stands for the number of items in the array of contributions.
 * @return The double representing the Contribution of the specific
 * parameter towards the overall rating of this response.
 */
public double getContribution(int number) {
    double contrib = 0;
    if(number >= 1 && number <= contributions.length) {
        contrib = contributions[(number-1)];
    }
    return contrib;
}

/**
 * Prints the information of this Response to a String.
 * <p>
 * It is in the following format:
 * <code>&lt;value[i]&gt; &lt;contribution[i]&gt; ... &lt;value[n]&gt; &lt;contribution[n]&gt; &lt;endValue&gt;
&lt;neutral&gt;</code></p>
 */
public String getPrintInfo() {
    String info = number + ",";
    for(int i=0; i<values.length; i++) {
        info = info.concat(values[i] + "," + contributions[i] + ",");
    }
    info = info.concat(endValue + "," + neutral);
    return info;
}

/**
 * Prints the information of this Response to a String, according to the format Vanhooff specified.
 * <p>
 * It is in the following format:
 * <code>&lt;value[i]&gt; &lt;contribution[i]&gt; ... &lt;value[n]&gt; &lt;contribution[n]&gt; &lt;endValue&gt;
&lt;neutral&gt;</code></p>
 *
 * @param MAX The maximum value of an individual attribute.
 */
public String getPrintInfoVHS(double MAX) {
    String info = number + ",";
    for(int i=0; i<values.length; i++) {
```

```
        info = info.concat((values[i]*MAX) + ";" + Math rint(contributions[i]*MAX*10)/10 + ";");
    }
    info = info.concat((endValue*MAX) + ";" + Math rint(neutral*MAX*10)/10);
    return info;
}
}
```

11.2.5 – UniNorms.Calculator

```
package UniNorms;
/**
 * The Calculator is there to make all necessary calculations to get from the
 * raw input to the estimated neutral value.
 * This is achieved by firstly calculating the alpha-value.
 * This alpha-value is fed into a formula, specified in the document accompanied
 * by the java-classes.
 *
 * @author Maarten Tijhof (275772mt@eur.nl)
 * @version 0.5 (2006-03-10)
 */
public class Calculator {

    /**
     * Constructor for objects of class Calculator.
     * <p>
     * It does nothing but instantiating an empty Calculator.
     * Further methodcalls must be done on the created instance.
     */
    public Calculator() {
        // do nothing
    }

    /**
     * The actual Calculator.
     * <p>
     * Takes the Response supplied and calculates an alpha-value first.
     * After this it will calculate the neutral-value and returns it.
     *
     * @param in The Response which is to be handled.
     * @see Response
     */
    public void calculate(Response in) {
        double alpha = calculateAlpha(in.getValues(), in.getEndValue());
        double neutral = calculateNeutralValue(alpha);
        in.setContributions(calculateContributions(in.getValues(), in.getEndValue(), alpha));
        in.setNeutral(neutral);
    }

    /**
     * Calculates the alpha-value.
     * <p>
     * Takes the supplied values and calculate the alpha.
     * For the algebra, please refer to the supplied formula-sheet.
     *
     * @param values The satisfactionlevels for the different aspects of the
     * product.
     * @param endValue The satisfactionlevel for the entire product.
     * @return The calculated alpha-value.
     */
    public double calculateAlpha(double[] values, double endValue) {
        double lnSum = 0;
        for(int i=0; i<values.length; i++) {
            double sLevel = values[i];
            double sLn = Math.log(sLevel/(1-sLevel));
            lnSum += sLn;
        }
        double n = values.length;
        double lnEnd = Math.log(endValue/(1-endValue));
        double alpha = (1/(n-1))*(-1*lnEnd+lnSum);
        return alpha;
    }
}
```

```
}

/**
 * Calculates the neutral-value.
 * <p>
 * Returns the calculated neutral-value by making calculations based on
 * the alpha.
 * For the algebra, please refer to the supplied formula-sheet.
 *
 * @param alpha The alpha-value.
 * @return The calculated neutral-value.
 */
public double calculateNeutralValue(double alpha) {
    double value = Math.exp(alpha)/(1+Math.exp(alpha));
    return value;
}

/**
 * Calculates all contributions of the seperate values.
 * It does so by doing the following calculation:
 * contribution[i] = value[i] - neutral
 * <p>
 * This is only a temporary solution.
 * Today (2005-01-27) I have worked out the formula to do the right calculations,
 * this will be implemented as soon as possible. (Yet, not today)
 * <p>
 * At last I've managed to get all formulas error-free, and thus I present to you:
 * The final implementation.
 * This could not have been true if it weren't for my tutor, drs. R Potharst.
 * My thanks fly out to you.
 * As of 2006-03-10 it is done.
 * For any detail of the applied algebra, please refer to the supplied formula-sheet.
 *
 * @param values The double-array containing all satisfactionlevels.
 * @param endValue The double representing the overall satisfaction.
 * @param alpha The calculated alpha value.
 * @return contributions The calculated contributions, in the same order as values.
 */
public double[] calculateContributions(double[] values, double endValue, double alpha) {
    double[] contributions = new double[values.length];

    double gs = Math.log(endValue/(1-endValue));
    double gsi = 0;
    double contr = 0;
    for(int i=0; i<contributions.length; i++) {
        gsi = Math.log(values[i]/(1-values[i]));
        contr = endValue-(1/(1+Math.exp(-1*(alpha+gs-gsi))));
        contributions[i] = contr;
    }
    return contributions;
}
}
```

11.2.6 – Utils.RandomCreator

```
package Utils;
import java.io.PrintWriter;
import java.io.BufferedWriter;
import java.io.FileWriter;

/**
 * The RandomCreator creates data for use in the {@link UniNorms} package.
 * <p>
 * It (at this moment) only has some pre-defined variables, as it didn't have any
 * other uses than the one it was used for: generating a random set.
 * This only happened once, but I included it anyway.
 * </p>
 * <p>
 * The standard-variables are:
```

```
* <ul>
* <li>FILENAME: data\random.csv<br>
* This is the file to which the results are written.</li>
* <li>MAXRESP: 1000<br>
* The amount of entries created. This is the number of questionees generated.</li>
* <li>MAXVAR: 20<br>
* The number of questions simulated, the overall-opinion not included.</li>
* <li>MAXNUM: 5<br>
* The maximum value the attributes can be given. Attributes being the questions.</li>
* </ul>
*
* @author Maarten Tijhof (275772mt@eur.nl)
* @version 0.1 (2006-03-22)
*/
public class RandomCreator
{
    private static final String FILENAME = "data\\random_data.csv";
    private static final int MAXRESP = 1000;
    private static final int MAXVAR = 20;
    private static final int MAXNUM = 5;

    /**
     * Generates the data.
     * <p>
     * Does so by taking MAXRESP*MAXVAR randoms within the numbers 1 and MAXNUM (inclusive) and writing
     * them in a NL-locale compatible CSV-file to the drive.
     * </p>
     */
    public RandomCreator() {
        PrintWriter out = null;
        try {
            out = new PrintWriter(new BufferedWriter(new FileWriter(FILENAME)), true);
        } catch (Exception e) {
            System.out.println("Something went horribly wrong whilst initiating writing the file " + FILENAME);
            System.out.println(" The returned error is: " + e);
        }
        try {
            for(int i=0; i<MAXRESP; i++) {
                for(int j=0; j<(MAXVAR+1); j++) {
                    out.print(new Double(Math rint(Math.random()*(MAXNUM-1)+1)).intValue()+",");
                }
                out.println();
            }
        } catch (Exception e) {
            System.out.println("Something went horribly wrong whilst writing to the file " + FILENAME);
            System.out.println(" The returned error is: " + e);
        }
    }
}
```

11.3 – Datasets

11.3.1 – Dataset A

Vragenlijst:

1. What is your opinion on the overall organization of this course?
2. How well have you been informed about the learning goals at the start of this course?
3. How well have you been informed about the examination at the start of this course?
4. Have you seriously prepared your: lectures
5. Have you seriously prepared your: workshops ("responsiecolleges")
6. "The content of this course is relevant for an education in <studie>."
7. Does this course require less/more study hours than the indicated number of credits?
8. How difficult is the content of this course?
9. Are the used methods of education appropriate, considering the learning goals?; lectures
10. Are the used methods of education appropriate, considering the learning goals?; workshops ("responsiecolleges")
11. To which extent have you been stimulated to self-study right from the start of the course?
12. Do you feel you have learned something useful as a result of this course?
13. What is your opinion on the books/course materials (see below) used in this course?;<book>.
14. How content are you with the information services rendered for this course through the Blackboard?
15. What is your opinion on the lectures of: <professor A>
16. What is your opinion on the lectures of: <professor B>
17. What is your opinion on the lectures of: <professor C>
18. How content are you with the possibilities for personal contact with: <professor A>
19. How content are you with the possibilities for personal contact with: <professor B>
20. How content are you with the possibilities for personal contact with: <professor C>
21. What is your overall opinion on this course?

11.3.2 – Dataset B

Vragenlijst:

1. What is your opinion on the overall organization of this course?
2. How well have you been informed about the learning goals at the start of this course?
3. How well have you been informed about the examination at the start of this course?
4. Have you seriously prepared your: lectures
5. Have you seriously prepared your: lecture with the video "A Soap Opera"

6. "The content of this course is relevant for an education in <studie>."
7. Does this course require less/more study hours than the indicated number of credits?
8. How difficult is the content of this course?
9. Are the used methods of education appropriate, considering the learning goals?; lectures
10. Are the used methods of education appropriate, considering the learning goals?; lecture with the video "A Soap Opera"
11. To which extent have you been stimulated to self-study right from the start of the course?
12. Do you feel you have learned something useful as a result of this course?
13. What is your opinion on the books/course materials (see below) used in this course? <book>.
14. What is your opinion on the books/course materials (see below) used in this course?; collegestof
15. How content are you with the information services rendered for this course through the Blackboard?
16. What is your opinion on the lectures of: <professor A>
17. What is your opinion on the lectures of: guestlecture: <lecturer A>
18. What is your opinion on the lectures of: guestlecture: <lecturer B>
19. What is your opinion on the lectures of: guestlecture: <lecturer C>
20. How content are you with the possibilities for personal contact with: <professor A>
21. What is your overall opinion on this course?

11.3.3 – Dataset C

Vragenlijst:

1. What is your opinion on the overall organization of this course?
2. How well have you been informed about the learning goals at the start of this course?
3. How well have you been informed about the examination during this course?
4. Have you seriously prepared the: plenary lectures (hoorcolleges)?
5. Have you seriously prepared the: "responsiecolleges"
6. "The content of this course is relevant for an education in <studie>."
7. Does this course require less/more study hours than the indicated number (6) of credits (1 erts = 28 uur)?
8. How difficult is the content of this course?
9. Are the used methods of education appropriate, considering the learning goals?; plenary lectures (hoorcolleges)
10. Are the used methods of education appropriate, considering the learning goals?; "responsiecolleges"
11. To which extent have you been stimulated to self-study right from the start of the course?
12. Do you feel you have learned something useful as a result of this course?

13. What is your opinion on the literature/course materials used in this course?;
overheads/transparancies
14. What is your opinion on the literature/course materials used in this course? articles on BB
15. What is your opinion on the literature/course materials used in this course? <book A>
16. What is your opinion on the literature/course materials used in this course? <book B>
17. What is your opinion on the literature/course materials used in this course? <book C>
18. How content are you with the information services rendered for this course through the
Blackboard?
19. What is your opinion on the lectures of: <professor A>
20. What is your opinion on the lectures of: <professor B>
21. What is your opinion on the lectures of: <professor C>
22. What is your opinion on the lectures of: <professor D>
23. What is your opinion on the lectures of: <professor E>
24. How content are you with; the possibilities for personal communication/contact with the
lecturers?
25. What is your overall opinion on this course?