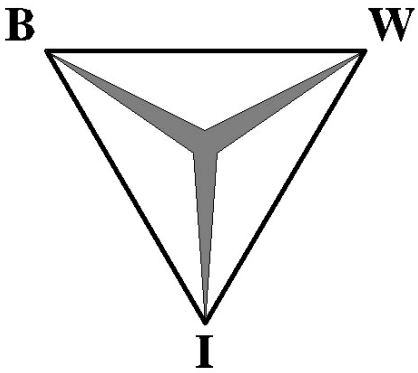


Verslag

NLP-optimale kalibratie van incomplete datamatrices

Generatie van intelligente feasible startwaarden

Wim Liu, VU/Bedrijfswiskunde en Informatica

onder begeleiding van

Prof. dr. A.E. Eiben, VU
drs. S.A. Pot, VU
drs. P.W.J.M. van Berkel, EIM BV

Voorwoord

Dit verslag is geschreven naar aanleiding van mijn afstudeerstage bij EIM BV te Zoetermeer. Deze stage is de afrondingsfase van mijn opleiding Bedrijfswiskunde en Informatica.

EIM richt zich op sociaal-economisch beleidsonderzoek voor overheden, beleidsinstanties en organisaties van het bedrijfsleven. EIM voert onderzoek uit over de hele breedte van het bedrijfsleven met als bijzondere specialisatie het midden- en kleinbedrijf (MKB). De onderzoeksprojecten beslaan een grote diversiteit aan onderwerpen. EIM BV bestaat uit units, één van de units is DataWarehousing.

Het stageonderzoek is uitgevoerd binnen de unit DataWarehousing. Binnen DataWarehousing worden voornamelijk projecten uitgevoerd die tot doel hebben gegevens van binnen en buiten het EIM te verbeteren en openbaar te maken, bijvoorbeeld via internet. De stage had betrekking op een belangrijk onderdeel van het beter maken van de data.

Mijn dank gaat uit naar mijn begeleiders van de VU, Prof. Dr. A.E. Eiben en drs. S.A. Pot voor het geven van nuttige tips. In het bijzonder wil ik mijn stagebegeleider drs. P. van Berkel bedanken voor alles wat hij mij heeft geleerd. Tot slot wil ik mijn vrienden bedanken voor de gegeven steun.

Wim Liu

Augustus 2004

Inhoudsopgave

1 Inleiding

De unit DataWarehousing van EIM BV heeft onder andere als taak het screenen en klaarmaken van complexe datasets voor onderzoeksdoeleinden en econometrische modellen. De sets worden van het CBS gekocht en deze sets bevatten naast ontbrekende waarden (zowel geheimgemaakt als echt onbekend) ook telfouten en afrondingsverschillen.[1] Deze datasets bestaan altijd uit een aantal niveaus, elk niveau bestaat uit variabelen en er is niet voor elke variabele een waarde bekend. Ook gelden er voor die variabelen simpele rekenregels waar de variabelen aan moeten voldoen. Maar dit is niet altijd het geval, dit komt door afrondingen, niet consistente gegevens en andere fouten. Om toch met de dataset te kunnen werken moet er een volledige consistente dataset zijn, dwz alle ontbrekende waarden moeten ingevuld zijn en deze moeten voldoen aan de rekenregels. Om al deze ontbrekende waarden ingevuld te krijgen wordt er gebruik gemaakt van de non-linear programming (NLP) techniek. Dit gebeurt in het programma SAS. Naast het schatten van de ontbrekende waarden wordt NLP ook gebruikt om (1) er voor te zorgen dat de wel gegeven data betreft deze zo min mogelijk afwijkt van de geleverde dataset, en die (2) rekentechnisch consistent is met alle bekende rekenregels en (domein)restricties. Voor het huidige probleem is het niet belangrijk om de werking en andere eigenschappen van NLP te onderzoeken.

1.1 Huidige oplossingsmethode

De data die EIM krijgt, zijn dus onvolledige data van het CBS. Om deze te kunnen gebruiken, worden eerst de nodige voorbewerking op de data uitgevoerd, zoals het afleiden uit voorgaande jaren en schatten door een materiedeskundige. Na deze stappen blijven er nog altijd missing waarden in de data over. Om ook deze te schatten worden de volgende stappen ondernomen:

1) Alle ontbrekende waarden worden met behulp van de rekenregels gecontroleerd of deze niet in de bijbehorende rekenregel de enige ontbrekende is. Bijvoorbeeld als variabele A geen waarde heeft en A + B = C is de bijbehorende rekenregel en B en C zijn wel bekend, dan kan A dus eenvoudigweg worden afgeleid. Deze stap wordt dan herhaald omdat er nu misschien andere rekenregels zijn, die in eerste instantie niet kunnen worden opgelost, maar nu wel, aangezien er in de vorige stap ontbrekende waarden zijn berekend. Bijvoorbeeld A + D = E, waar alleen E bekend is, aangezien A in de vorige stap is berekend, kan D nu ook worden afgeleid. Deze stap stopt als er geen ontbrekende waarden op deze manier kan worden afgeleid.
2) Alle missing waarden die niet op de bovenstaande manier kunnen worden berekend, worden nu op nul gezet.

De ontbrekende waarden worden op nul gezet omdat NLP niet met ontbrekende waarden kan werken. De hierboven verkregen dataset wordt als startwaarde gebruikt

[1] Als er een nul in de dataset voorkomt, dan is die nul niet ontbrekend! Een nul stelt dus ook een waarde voor. Een ontbrekende waarde wordt in de dataset met een '.' (punt) aangeven.

voor de NLP techniek. NLP maakt dan gebruik van de rekenregels om de dataset consistent te maken.

1.2 Probleemformulering

Hoewel de huidige oplossingsmethode werkt, is er toch een aantal inhoudelijke en praktische vragen die onbeantwoord zijn. De belangrijkste vraag betreft het **startwaarde probleem**, dit wil zeggen het genereren van een dataset die *vooraf* zodanig is ingevuld dat hij kan dienen als zogenaamde *feasible* oplossing om het NLP-optimalisatieproces mee te beginnen. Nu wordt er dus een nul ingevuld voor de ontbrekende waarden, omdat deze binnen het grensgebied van de rekenregels ligt. Om het NLP-optimalisatieproces te starten, moeten er aannemelijke waarden zijn die binnen het grensgebied liggen. Hoewel de bovenstaande startwaarde set wel voldoet, is er toch een aantal gevallen bekend waar het niet goed gaat.[2] Dit is hoogstwaarschijnlijk te wijten aan de vele nullen die vooraf zijn ingevuld. Er wordt daarom gezocht naar oplossingen die betere waarden geven dan een nul.
De vooraf ingevulde waarden die dichter bij de oplossing ligt, heeft natuurlijk de voorkeur boven de nullen. Deze 'betere' waarden zorgen namelijk voor een snellere berekening en waarschijnlijk ook voor een betere oplossing.
Naast dit probleem, speelt er ook een aantal deelproblemen een rol, die het oplossen van het startwaarde probleem bemoeilijkt. Deze deelproblemen worden zo nu en dan wel genoemd, maar dit verslag behandelt voornamelijk alleen het startwaarde probleem:

- Het CBS gaat niet of niet altijd goed om met de aard van de gegevens. Het onderscheid tussen missing values (niet waargenomen of geheim) en echt nul lijkt niet altijd correct te worden weergegeven. Het vaststellen van de aard van de data is belangrijk omdat dit bijdraagt aan de betrouwbaarheid van de oplossing. Een echte nul verandert namelijk niet en een '.' (ontbrekend) wel.
- Er is reden te veronderstellen dat de huidige oplossingsroutine, het onthullen, gevoelig is voor de volgorde van toepassen van rekenregels. De gegevens die we hebben bevatten namelijk allerlei fouten. Als er eerst met deze foute gegevens wordt gewerkt, dan worden de fouten als het ware doorgegeven aan andere variabele.
- Tijdens de NLP-fase worden sommige variabelen erg vaak gebruikt, deze variabelen worden dan door meerdere rekenregels 'aangeroepen'. Het is niet bekend wat voor invloed deze zelfweging heeft op de oplossing van NLP.

1.3 Doelstelling

Om het probleem (geschikte startwaarde vinden) te kunnen oplossen, is het verstandig om meer over de achtergrond en het proces te weten te komen. Daarom wordt de doelstelling als volgt gedefinieerd:

[2] Of een dataset goed is of niet wordt bepaald door de materiedeskundige. Deze bekijkt dan de dataset en beoordeeld deze.

Beschrijf alle relevante onderwerpen die kunnen bijdragen bij het oplossen van het probleem. Dus het proces, opbouw van de data, opbouw van de rekenregels etc.

1.4 Structuur van dit verslag

Dit verslag is als volgt opgebouwd. In hoofdstuk 2 zal eerst worden uitgelegd hoe de data er uitzien en worden structuur en eigenschappen van de data besproken.
In hoofdstuk 3 worden de rekenregels toegelicht. Ook wordt er in dit hoofdstuk uitgelegd wat er gebeurt als de rekenregels op de data worden toegepast.
In hoofdstuk 4 wordt een aantal technieken uit de literatuur besproken. Deze technieken kunnen niet gebruikt worden voor het probleem, de bedoeling is te zoeken naar elementen die mogelijk bruikbaar zijn.
In hoofdstuk 5 wordt vervolgens een aantal technieken besproken die wel in staat zijn een oplossing te leveren. Er wordt vaak eerst een algemeen geval behandeld gevolgd door een voorbeeld. Deze voorbeelden worden toegelicht met namen/labels. In de meeste gevallen is het handig als de omschrijving van de namen bekend is, deze omschrijving staat in Bijlage 1.
In hoofdstuk 6 zal uiteindelijk een algemene aanbeveling worden gegeven.

2 Formulering datastructuur

In dit hoofdstuk zal de structuur en eigenschappen van de data worden behandeld. In paragraaf 2.1 zal de data op een wiskundige wijze worden weergegeven. Vervolgens zal in paragraaf 2.2 de data in andere vormen worden weergegeven.

2.1 De data vanuit een wiskundig perspectief

De data die we krijgen bestaan uit waarden van allerlei bedrijven van Nederland. Laat nu het volgende gedefinieerd zijn:

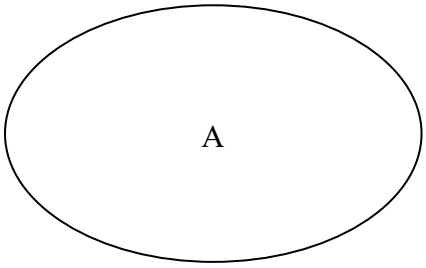
A : = alle bedrijven van Nederland
B_j : = alle bedrijven in bedrijfsgroep j met j = 1, ..., J
C_k : = alle bedrijven in grootteklasse k met k = 1, ..., K
D : = verzameling van alle statistics
x_{mi} : = de waarde van bedrijf m voor statistic i, $i \in D$ en $m \in A$

Voorbeeld:
A : = {ABN-AMRO, café Bruinsma, loodgieter de Pijp, bakker Blank, ...}
B_j : = {horecasector, bouwnijverheidsector, loodgieter, ...}
C_k : = {# werknemers = 0, # werknemers =1 t/m 9, ..., # werknemers >=100}
D : = {omzet, winst, voorzieningen, ...}
x_{mi} : = de winst (i) van bakker Blank (m)

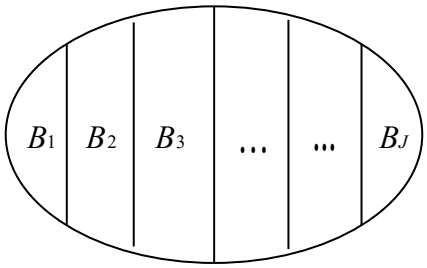
dan geldt het volgende:
$B_j \subseteq A$ en $C_k \subseteq A$
$x_{jki} = \sum_m x_{jkim}$ met $m \in B_j \cap C_k$ en $i \in D$

met x_{jki} de variabele die de totale waarde bevat voor alle bedrijven in bedrijfsgroep (BGR) j en grootteklasse (GKL) k van statistic (VAR) i.
Voorbeeld:
x_{jki} : = de omzet (i) van alle loodgieters (j) bedrijven met aantal werknemers 1 t/m 9 (k)

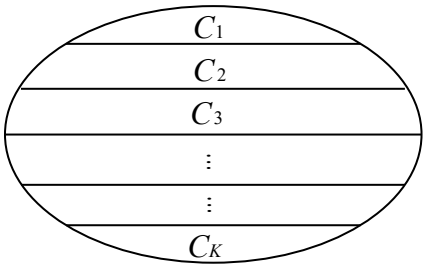
met andere woorden, bedrijfsgroep j is een deelverzameling van de Nederlandse economie, ook grootteklasse k is een deelverzameling van de Nederlandse economie. Dit betekent: elk element van B is ook een element van A, en ook elk element van C is een element van A. De Nederlandse economie wordt dus onderverdeeld in bedrijfsgroepen en grootteklassen. In Figuur 2-1 t/m Figuur 2-4 staan de diagrammen van deze verzamelingen.
In Figuur 2-5 staat een afbeelding van $B_j \cap C_k$ voor j=2 en k=2. Alleen de bedrijven die in het dubbel gearceerd stuk voorkomen worden gebruikt.



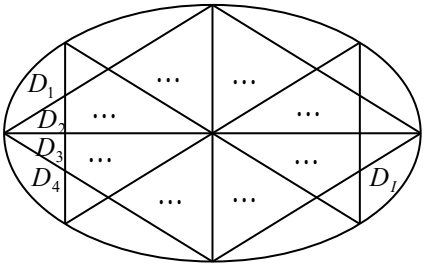
Figuur 2-1 gehele economie van Nederland



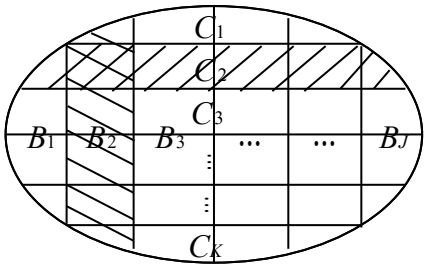
Figuur 2-2 gehele economie onderverdeeld in BGR



Figuur 2-3 gehele economie onderverdeeld in GKL



Figuur 2-4 gehele economie onderverdeeld in VAR



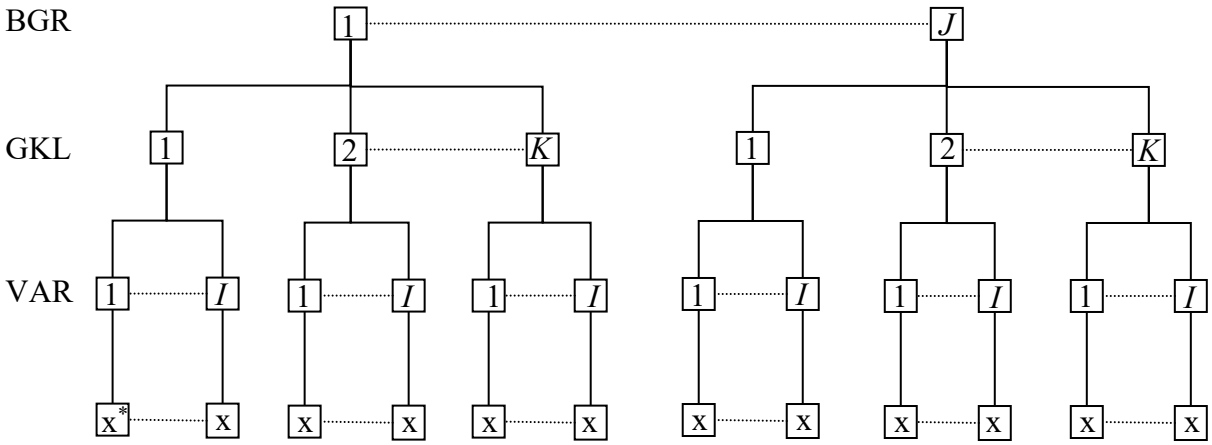
Figuur 2-5 doorsnede van B_2 en C_2 [3]

2.2 De data vanuit een niet wiskundig perspectief

Na de wiskundige formulering van de data worden deze nu weergegeven met behulp van bomen en tabellen, beide worden weergegeven met indexnummers. Deze staan respectievelijk in Figuur 2-6 en Tabel 2-1. In Figuur 2-6 is te zien dat de data uit drie niveaus bestaan (sommigen kunnen er meer hebben). Namelijk de bedrijfsgroep (BGR), de grootteklasse (GKL) en de statistics (VAR). Onder aan de boom staat de waarde voor de desbetreffende variabele, deze waarde kan dus ook onbekend zijn. In Bijlage 2 wordt er ook een boom weergegeven met de echte labels van een bepaalde batch in plaats van een index. Zie Bijlage 1 voor de betekenis van een label.

[3] Door de vorm van Figuur 2-5 wordt de indruk gewekt dat B_1 en C_1 geen doorsnede met elkaar hebben, dit is niet waar. Elke verzameling van B heeft een doorsnede met elke verzameling van C. Dus $\forall j$ en $\forall k$ geldt $\exists m \in B_j \cap C_k$.

In Tabel 2-1 wordt dezelfde data in een tabel vorm gezet. Een waarde heeft altijd drie ‘labels’. x^* heeft dus de label BGR, GKL en VAR, met het indexnummer 1. Of ook, BGR 1 met GKL 1 en VAR 1 heeft als waarde x^*.



Figuur 2-6 Boomstructuur van de data met ‘x’ een waarde voor een variabele

BGR	GKL	VAR 1	...	VAR *i*	...	VAR *I*
1	1	x^*		x		x
	...	x		x		x
	k	x		x		x
	...	x		x		x
	K	x		x		x
...	...	...	...	...	...	...
j	1	x		x		x
	...	x		x		x
	k	x		x		x
	...	x		x		x
	K	x		x		x
...	...	...	...	...	...	...
J	1	x		x		x
	...	x		x		x
	k	x		x		x
	...	x		x		x
	K	x		x		x

Tabel 2-1 tabelstructuur van de data met ‘x’ een waarde voor een variabele

Uit bovenstaande figuren kunnen het volgende worden afgeleid:
- Aantal waarden van BGR *j* is gelijk aan *K* * *I*
- Aantal waarden van GKL *k* is gelijk aan *I*
- Aantal waarden van VAR *i* is gelijk aan *J* * *K*
- Aantal waarden van BGR *j* in combinatie met GKL *k* is gelijk aan *K* * *I*

3 Formulering rekenregels

In dit hoofdstuk zullen de gebruikte rekenregels worden behandeld. In paragraaf 3.1 zal eerst een algemene formulering worden gegeven. Vervolgens wordt in paragraaf 3.2 een voorbeeld gegeven van de rekenregels. Tot slot wordt in paragraaf 3.3 een functie gegeven die gebruikt wordt om de afwijking tussen de originele data en de NLP-data te minimaliseren.

3.1 Algemene formulering rekenregels

Laat x_{jki} de variabele zijn die de waarde bevat voor bedrijfsgroep (BGR) *j* en grootteklasse (GKL) *k* van statistic (VAR) *i*. Met *j*=1,..., *J* , *k*=1,..., *K* en *i*=1,..., *I*. Op elk van deze drie groepen zijn rekenregels gedefinieerd. De rekenregels voor de bedrijfsgroepen worden als volgt gegeven door:

1) $x_{b_1 ki} = \sum_{j=1,...,J,\, j \neq b_1} w_{jki} \cdot x_{jki}$ voor *k* = 1,..., *K* , *i* = 1,..., *I* en *w* = -1, 0, 1

2) ....

.. ..

N_b) $x_{b_{N_b} ki} = \sum_{j=1,...,J,\, j \neq b_{N_b}} w_{jki} \cdot x_{jki}$ voor *k* = 1,..., *K* , *i* = 1,..., *I* en *w* = -1, 0, 1

met N_b het aantal rekenregels voor BGR, deze rekenregels worden gedefinieerd als BGR_RULES.

die van de grootteklassen worden gegeven door:

1) $x_{jg_1 i} = \sum_{k=1,...,K,\, k \neq g_1} w_{jki} \cdot x_{jki}$ voor *j* = 1,..., *J* , *i* = 1,..., *I* en *w* = -1, 0, 1

2) ....

.. ..

N_g) $x_{jg_{N_g} i} = \sum_{k=1,...,K,\, k \neq g_{N_g}} w_{jki} \cdot x_{jki}$ voor *j* = 1,..., *J* , *i* = 1,..., *I*, en w = -1, 0, 1

met N_g het aantal rekenregels voor GKL, gedefinieerd als GKL_RULES.

en die van de statistics:

1) $x_{jkv_1} = \sum_{i=1,...,I,\, i \neq v_1} w_{jki} \cdot x_{jki}$ voor *j* = 1,..., *J* , *k* = 1,..., *K* en *w* = -1, 0, 1

2) ....

.. ..

N_v) $x_{jkv_{N_v}} = \sum_{i=1,...,I,\, i \neq v_{N_v}} w_{jki} \cdot x_{jki}$ voor *j* = 1,..., *J* , *k* = 1,..., *K* en *w* = -1, 0, 1

met N_v het aantal rekenregels voor VAR, gedefinieerd als VAR_RULES.

$-\infty < x_{jki} < \infty$[4]

Aantal variabelen is $J * K * I$.

Afhankelijk van de gebruikte dataset worden er andere sets van rekenregels gebruikt. Deze sets ontstaan door potentiële rekenregels te onderzoeken. Er wordt dan onderzocht of alle variabelen die in de rekenregel voorkomen, ook voorkomt in de dataset, zo nee dan wordt deze rekenregel niet geselecteerd. Deze rekenregels waren ooit door een inhoudelijke deskundige vastgesteld en staan in de metadata. Rekenregels die worden geselecteerd moeten dan ook worden gebruikt om ontbrekende variabelen te berekenen/schatten. Deze rekenregels zijn strikt lineair; alleen optellen voor de BGR_RULES en GKL_RULES en/of aftrekken voor de VAR_RULES is toegestaan. Andere rekenkundige operaties zijn er niet. Het aantal keren dat een rekenregel wordt gebruikt wordt in de onderstaande formule gegeven:

Totaal aantal rekenregels voor BGR is: $K * I * N_b$
Totaal aantal rekenregels voor GKL is: $J * I * N_g$
Totaal aantal rekenregels voor VAR is: $J * K * N_v$

3.2 Voorbeeld rekenregels

De rekenregels voor bijvoorbeeld van batch 9 voor de bedrijfsgroepen worden als volgt weergegeven gebruik makend van de BGR_RULES (zie Bijlage 1 voor betekenis van de indexen):

$$x_{4ki} = x_{5ki} + x_{6ki} \tag{1.1}$$

$$x_{1ki} = x_{2ki} + x_{3ki} + x_{4ki} + x_{7ki} + x_{8ki} + x_{9ki} + x_{10,k,i} + x_{11,k,i} + x_{12,k,i} + x_{13,k,i} + x_{14,k,i} \tag{1.2}$$

voor $k = 1,..., 7$ en $i = 1,..., 83$
die van de grootteklassen worden met de GKL_RULES als volgt weergegeven:

$$x_{j2i} = x_{j1i} + x_{j7i} \tag{1.3}$$

$$x_{j6i} = x_{j3i} + x_{j4i} \tag{1.4}$$

$$x_{j5i} = x_{j2i} + x_{j3i} + x_{j4i} \tag{1.5}$$

voor $j = 1,..., 14$ en $i = 1,..., 83$
en tot slot worden de statistics rekenregels met de VAR_RULES weergeven:

[4] Afhankelijk van de betekenis en definitie van de variabele kan deze ook uitsluitend positieve waarden aannemen. Voorbeeld: afschrijving.

$$x_{jk66} = x_{jk64} + x_{jk65} \tag{1.6}$$
$$x_{jk21} = -x_{jk28} + x_{jk29} \tag{1.7}$$
$$x_{jk22} = -x_{jk32} + x_{jk35} \tag{1.8}$$
$$x_{jk67} = x_{jk37} + x_{jk47} + x_{jk64} + x_{jk65} + x_{jk74} \tag{1.9}$$
$$x_{jk80} = x_{jk58} + x_{jk77} \tag{1.10}$$
$$x_{jk79} = x_{jk57} + x_{jk78} \tag{1.11}$$
$$x_{jk82} = x_{jk53} \tag{1.12}$$
$$x_{jk2} = x_{jk34} + x_{jk38} + x_{jk82} \tag{1.13}$$
$$x_{jk41} = x_{jk80} \tag{1.14}$$
$$x_{jk39} = x_{jk79} \tag{1.15}$$
$$x_{jk6} = x_{jk2} + x_{jk4} \tag{1.16}$$
$$x_{jk46} = -x_{jk39} + x_{jk41} \tag{1.17}$$
$$x_{jk55} = -x_{jk46} + x_{jk59} + x_{jk75} + x_{jk76} + x_{jk81} \tag{1.18}$$

voor i = 1,..., 14 , j = 1,..., 7.
-5000 < x_{jki} < 120000

Aantal variabelen N is j * k * i = 14 * 7 * 83 = 8134.

De hier gegeven rekenregels zijn alle rekenregels die in batch 9 worden gebruikt. Het aantal variabelen bij andere batches kan wel oplopen tot J * K * I = 100 * 8 * 100 = 80000. Het aantal keren dat (1.1) wordt toegepast is dus 7 * 83 = 581 keer. Het aantal rekenregels dat ontstaat uit bedrijfsgroepen regels is 581 * 2 = 1162. De formule voor het aantal rekenregels voor BGR is # GKL * # VAR * # rekenregels van BGR. In totaal worden er 5922 rekenregels gebruikt. De berekening wordt hieronder getoond:

BGR	7	*	83	*	2	=	581	*	2	= 1162
GKL	14	*	83	*	3	=	1162	*	3	= 3486
VAR	7	*	14	*	13	=	98	*	13	= 1274
Totaal										5922

Uit bovenstaande berekening blijkt dat met ruim 20 algemene rekenregels er ongeveer 6000 rekenregels afgeleid kunnen worden. Deze 6000 rekenregels worden vervolgens gebruikt om 8000 variabelen te berekenen. Tijdens de NLP-fase moeten alle variabelen aan al deze 6000 rekenregels voldoen.

Door de betekenissen van de variabelen voor de bedrijfsgroepen en grootteklassen staat er aan de linkerkant van de vergelijking altijd een subtotaal dan wel een totaal van die groep. Bedrijfsgroepen en grootteklassen vormen namelijk categorische classificaties. Als voorbeeld wordt weer batch 9 genomen, x_{1ki} is het totaal van de bedrijfsgroepen voor een bepaalde k en een bepaalde i en x_{j5i} is het totaal van de grootteklassen voor

een bepaalde *j* en een bepaalde *i*. In tegenstelling tot de bovenstaande twee groepen kan er geen totaal worden uitgerekend voor de statistic-groep. Hier worden namelijk verschillende eenheden gebruikt waardoor het uitrekenen van een totaal voor de VAR-groep niet mogelijk is, anders worden er appels met peren met elkaar vergeleken. Hoewel er dus niet gesommeerd kan worden over alle *i*'s, kan er wel de som worden uitgerekend voor elke *i* apart. Zie Bijlage 3 voor een schematische weergave.

In feite is er geen optimaliteitscriterium voor het berekenen van een startwaardeset, er wordt dus geen functie geminimaliseerd dan wel gemaximaliseerd, want met de gegeven rekenregels kan dit ook niet. De bedoeling is dat de x_{jki} voor alle rekenregels en voor alle *j*, *k* en *i* kloppen. Dus een variabele moet in alle richtingen exact kloppen of kloppend gemaakt worden met de bijbehorende rekenregels. Dit is een eis die aan NLP wordt gesteld.

Nadat alle ontbrekende x_{jki} geschat zijn, worden alle gegevens, inclusief de bekende waarden, aan NLP aangeboden. De rekenregels die aan NLP worden aangeboden zien er in matrix vorm, voor de grootteklassenregels, als volgt uit:

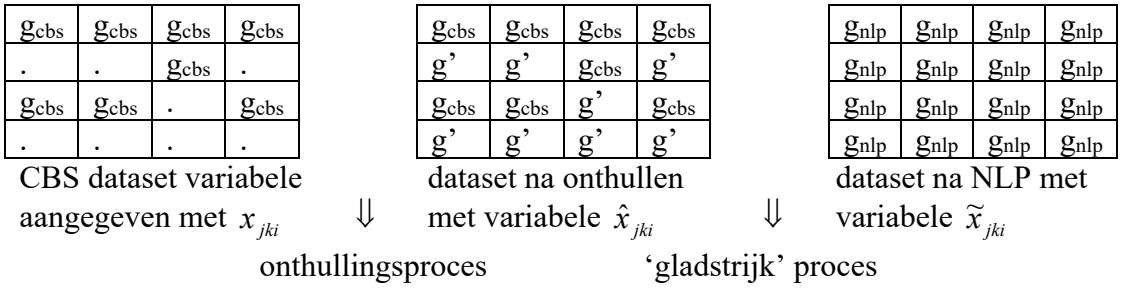
x_{j1i}	x_{j2i}	x_{j3i}	x_{j4i}	x_{j5i}	x_{j6i}	x_{j7i}
1	-1	0	0	0	0	1
0	0	1	1	0	-1	0
0	1	1	1	-1	0	0

Immers (1.3) kan ook herschreven worden als $0 = x_{j1i} + x_{j7i} - x_{j2i}$ en (1.4) als $0 = x_{j3i} + x_{j4i} - x_{j6i}$ deze staan respectievelijk in rij 2 en rij 3 van de matrix. Want $0 = x_{j1i} + x_{j7i} - x_{j2i}$ kan ook herschreven worden als $0 = 1 * x_{j1i} + 0 * x_{j3i} + 0 * x_{j4i} + 0 * x_{j5i} + 0 * x_{j6i} + 1 * x_{j7i} - 1 * x_{j2i}$ en deze staat dus in rij 2. Een soortgelijke matrix wordt ook van de andere groepen gemaakt. Er dient opgemerkt te worden dat er altijd één negatieve term in de vergelijking zit na het herschrijven van de rekenregels voor BGR en GKL. Voor de VAR regels kunnen er na het herschrijven ook twee of meer negatieve termen in de vergelijking voorkomen, dit betreft dan saldo posten, zie bijvoorbeeld (1.7). In (1.7) is er al een negatieve term in de rekenregel, na het herschrijven komt er dus nog eentje bij. Dergelijke rekenregels bemoeilijken het proces van het kloppend maken van de data. Want als er twee of meer variabele in zo'n rekenregel onbekend zijn, dan kunnen die variabelen theoretisch gezien een oneindig aantal waarden aannemen.

3.3 Doelfunctie

Om toch de dataset na NLP te vergelijken met de CBS dataset, wordt er gebruik gemaakt van de minimale kwadratensom. Voordat er naar deze functie wordt gekeken, wordt er nog voor de duidelijkheid heel kort het proces beschreven.
In principe kunnen er drie fases van de dataset worden onderscheiden. Namelijk 1) dataset van het CBS (x_{jki}), 2) dataset na het schatten van de ontbrekende waarden ($\hat{x}_{jki}$)

en 3) een dataset na NLP ($\tilde{x}_{jki}$). Zie Figuur 3-1 voor een schematische weergave van het proces. Dataset van het CBS komt dus eerst in het onthullingsproces, waar de ontbrekende waarden worden onthuld. De dataset na het onthullen komt dan in het 'gladstrijk' proces, waar de dataset consistent wordt gemaakt.



Figuur 3-1 schematische weergave van begin dataset tot eind dataset

Tijdens het onthullingsproces worden alleen de ontbrekende waarden afgeleid. Hier speelt de volgorde van het toepassen van een rekenregel een rol. Bij consistente data maakt het niet uit welke rekenregel eerst wordt toegepast om een variabele uit te rekenen. Maar de gegevens die we hebben, bevatten allerlei fouten waardoor het dus wel uitmaakt welke rekenregel eerst wordt toegepast voor de onthulling. Als er fouten in een variabele zitten, en deze wordt als eerste gebruikt dan worden de fouten als het ware doorgegeven aan andere variabelen. Er wordt namelijk niet in één keer alle ontbrekende waarden afgeleid. Zie Bijlage 4 om te zien hoe ontbrekende waarden worden afgeleid. Dus het op een slimme (lees optimale) manier toepassen van de rekenregels is essentieel voor de berekende waarden. Een theorie voor het toepassen van de volgorde van de regenregels is nog niet bedacht.

Bij het 'gladstrijk' proces is bekend dat sommige variabelen vaker worden gebruikt. Dit is ook te zien aan de rekenregels. Uit de rekenregels blijkt bijvoorbeeld dat $x_{1,5,i}$ is af te leiden uit (1.2),

$$x_{1,5,i} = x_{2,5,i} + x_{3,5,i} + x_{4,5,i} + x_{7,5,i} + x_{8,5,i} + x_{9,5,i} + x_{10,5,i} + x_{11,5,i} + x_{12,5,i} + x_{13,5,i} + x_{14,5,i}$$

maar deze is ook af te leiden uit (1.5).

$$x_{1,5,i} = x_{1,2,i} + x_{1,3,i} + x_{1,4,i} \qquad \text{voor i=1,..., 83}$$

Een variabele wordt dus als het ware meerdere keren gebruikt. Sommige variabelen worden dus vaker gebruikt dan anderen. Vastgesteld kan worden dat deze variabelen belangrijk zijn voor de procedures. Immers, als deze door zoveel rekenregels worden gebruikt, dan heeft het aanpassen van een dergelijke variabele invloed op veel andere variabelen.
Tijdens dit proces moeten ook alle rekenregels kloppen. Voor het voorbeeld met batch 9 zijn dit 5922 rekenregels.

Om er voor te zorgen dat de dataset na NLP zo min mogelijk afwijkt van de CBS dataset, wordt er dus gebruik gemaakt van de minimale kwadratensom. Deze wordt als volgt weergegeven:

min $\sum_j \sum_k \sum_i (\hat{x}_{jki} - \tilde{x}_{jki})^2$ voor alle gegeven $\hat{x}_{jki}$ (dit is exclusief de onthulde waarden, omdat dat verschil niet berekend kan worden)

De minimale kwadratensom wordt genomen over $\hat{x}_{jki}$ en $\tilde{x}_{jki}$, maar *niet* over de waarden die tijdens het onthullingsproces zijn afgeleid.
De aan NLP aangeboden dataset en rekenregels worden dus over deze functie geoptimaliseerd, maar zó dat de data ook aan de rekenregels voldoen.

4 Literatuuronderzoek

In dit hoofdstuk wordt een aantal technieken uitgelegd voor het berekenen/schatten van ontbrekende waarden. Deze technieken gaan er van uit dat de primaire data beschikbaar zijn. Hoewel wij daar niet over beschikken is het interessant om te onderzoeken of er geen bruikbare elementen zijn. In paragraaf 4.1 worden eerst drie gevallen uitgelegd die bij ontbrekende waarden kunnen worden onderscheiden. In paragraaf 4.2 tot en met paragraaf 4.4 worden vervolgens de technieken voor het schatten van de missing waarden uitgelegd. In deze paragrafen worden ook uitgelegd wat er gebeurt als deze technieken toch worden toegepast. Tot slot wordt in paragraaf 4.5 een conclusie getrokken van deze technieken.

4.1 Drie gevallen van missing value

Bij het toepassen van een rekentechniek wordt er altijd van uitgegaan dat de data voldoen aan bepaalde aannames. Bij ontbrekende waarden worden er over het algemeen drie soorten mechanismen onderscheiden.

1) Het eerste mechanisme is missing completely at random (MCAR). De ontbrekende waarde voor een variabele Y zijn MCAR als de kans op een ontbrekende waarde voor Y niet gerelateerd is aan de waarde van Y zelf of aan de waarden van andere variabelen in de dataset. Er is dus geen enkele samenhang tussen de waarden[3], [10].
2) Het tweede mechanisme is missing at random (MAR). De kans dat de waarde van Y ontbreekt, kan afhangen van andere variabelen, maar deze is onafhankelijk van de waarde van Y. Bijvoorbeeld de kans dat er bij een enquête het geslacht niet wordt ingevuld, kan afhangen van andere variabelen, bijvoorbeeld van leeftijd of opleiding. Maar deze hangt niet af van het geslacht zelf. De kans dat een mannelijke respondent een vraag niet beantwoord is gelijk aan de kans dat een vrouwelijke respondent die vraag ook niet beantwoord. Er is dus uitsluitend samenhang met andere variabelen.
3) Het laatste mechanisme betreft dan gegevens die niet voldoen aan de voorwaarde van MAR, dit betreft waarden die ontbreken omwille van hun waarde zelf. Deze worden not missing at random (NMAR) genoemd. Dit doet zich bijvoorbeeld voor bij een vraag naar het inkomen waar de hogere inkomens de vraag ontbeantwoordt laten. Deze gevallen zijn moeilijker hanteerbaar dan MAR. Hier is er dus sprake van samenhang met zichzelf

Voor het toepassen van imputation moet de dataset of voldoen aan de eisen van MAR of voldoen aan de eisen van MCAR. Als de dataset niet voldoet aan één van deze twee eisen kan het toepassen van deze technieken tot rare resultaten leiden. Dit kan tot onjuiste conclusies leiden.

4.2 Multiple Imputation (MI)

Een veelgebruikte techniek om missing waarden te schatten is imputation. Een voordeel van imputation is dat bij een op imputation gebaseerde techniek de 'lege velden' opgevuld worden met aannemelijke waarden om een complete dataset te creëren, die op zijn beurt weer geanalyseerd kan worden met standaard technieken. Een tweede voordeel is dat het maken van aannemelijke waarden maar een keer hoeft te gebeuren. Een nadeel van imputation is dat afleidingen gebaseerd op de geïmputeerde data geen rekening houdt met de onzekerheid die de ontbrekende waarden met zich brengen. Als de geïmputeerde data behandeld wordt alsof deze de 'echte' data zijn, dan wordt er een vertekend beeld gegeven van de data.

Er zijn eigenlijk twee soorten 'takken' van imputation, namelijk single imputation en multiple imputation. Het verschil is dat bij MI een waarde meerdere keren wordt geïmputeerd, terwijl dat bij een 'single' imputation slechts één keer gebeurt. Met MI wordt het probleem van de onzekerheid die ontstaat bij het schatten van ontbrekende waarden gecorrigeerd. Er zijn verschillende technieken om te imputeren, grofweg kunnen deze verdeeld worden in twee groepen.

1) De groep van methodes waar echte waarden worden geïmputeerd. Ontbrekende data kunnen dan bijvoorbeeld vervangen worden door historische gegevens.
2) Hier zijn de imputation methodes gebaseerd op expliciete modellen. Enkele hiervan zijn:
   -Gemiddelde imputation: vervang alle ontbrekende gegevens met het gemiddelde van de aanwezige gegevens. Hier wordt de standaard fout sterk onderschat.
   -Regressie imputation: voorspel ontbrekende gegevens met regressie. Imputeer elke variabele op basis van andere variabelen in het model. Deze methode produceert dan bevooroordeelde schattingen.

In multiple imputation wordt er gebruikt gemaakt van de bestaande gegevens om de ontbrekende gegevens te voorspellen. De geschatte waarden worden imputation genoemd en vervangen de ontbrekende gegevens. Zodoende wordt er een volledige dataset geleverd die de 'geïmputeerde dataset' wordt genoemd. Dit proces wordt meerdere keren herhaald, er ontstaan dan meerdere 'geïmputeerde datasets'. Stel m is het aantal sets dat gemaakt worden. Als m=5 dan wordt er van elke ontbrekende waarde van de data 5 keer een imputation gemaakt[5]. Deze 5 geïmputeerde sets zullen onderling van elkaar verschillen. Om deze datasets te gebruiken wordt er op elke 'geïmputeerde dataset' een standaard statistische analyse uitgevoerd. De resultaten van elke dataset worden dan gecombineerd om tot een algemene analyse te komen [9].

De bedoeling van imputation is niet de waarden van de ontbrekende data te gokken. De bedoeling is juist een geïmputeerde dataset te maken die de totale variabiliteit en relatie

[5] Voor de meeste toepassing zijn 3-5 imputaties genoeg om tot zeer goede resultaten te komen. Deze kan berekend worden met de formule $\left(1+\frac{\gamma}{m}\right)^{-1}$, met γ de fractie missing. Deze formule wordt gegeven door Rubin [6].

met andere variabelen behoudt. Voor deze techniek wordt er aangenomen dat de waarden aan de MAR voldoen.

4.2.1 Toepassing multiple imputation bij EIM

Multiple imputation maakt gebruik van de missing data patroon. Van elk voorkomend missing data patroon wordt er bijgehouden hoe vaak deze voorkomt [8]. Op basis van deze patronen wordt er een model gemaakt, die de missing waarden kan schatten. Als dit wordt toegepast in de situatie van EIM, dan kan een tabel er uit zien als Tabel 4-1:

BGR	GKL	VAR1	VAR2	...	...	VAR*n*
490	1 10 13 15	10 50 70 10	20 . 100 50	45 36 125 44	. 11 99 50	2 3 12 7
512	1 10 13 15	. . 30 .	12 5 38 21	14 16 50 20	. . . .	. . 22. .
enz	enz					

Tabel 4-1 mogelijke data van EIM

De bijhorende missing data patroon ziet er als volgt uit, als MI wordt toegepast per BGR met ‘x’ waargenomen en ‘.’ missing:

Group 490	VAR1	VAR2	...	...	VAR*n*	Frequentie
1	x	x	x	x	x	2
2	x	x	x	.	x	1
3	x	.	x	x	x	1

Group 512	VAR1	VAR2	...	...	VAR*n*	Frequentie
1	x	x	x	.	x	1
2	.	x	x	.	.	3

Tabel 4-2 missing data patroon

Hier worden de verschillende GKL’s met elkaar vergeleken. Kijken we naar de betekenis van een GKL dan zien we dat waarden die gebaseerd zijn op 1 werknemer gecombineerd worden met waarden die gebaseerd zijn op 1 t/m 10 werknemers en waarden die gebaseerd zijn op het totale aantal werknemers. Dit betekent dat de verschillende grootteklassen vergelijkbaar zijn, wat natuurlijk niet klopt. Waarden die gebaseerd zijn op 1 t/m 10 werknemers kan nou eenmaal niet vergeleken worden met waarden die gebaseerd zijn op 1 werknemer. Er wordt dus geen rekening gehouden met de rekenregels, wat op zich wel logisch is, want daar is deze techniek niet voor gemaakt.

Hetzelfde kan ook beredeneerd worden als de verschillende BGR’s met elkaar worden vergeleken. Als de BGR’s met elkaar worden gecombineerd, dan wordt er in feite gezegd dat de bedrijfsgroep metselaars dezelfde kosten en dergelijke hebben als een

andere bedrijfsgroep, bijv bouwvakkers, dit klopt ook niet. Er zijn geen bruikbare elementen gevonden om het huidige probleem (deels) op te lossen.

4.3 Case deletion

Case deletion staat ook bekend als case analyse. Deze komt voor in alle statistische pakketten en is een standaard methode in veel programma's. Deze methode verwijdert simpelweg ontbrekende waarden van één of meerdere variabelen in de analyse.
De voordelen van deze benadering zijn 1) simpel, standaard statistische analyse kunnen hier worden toegepast zonder enige aanpassingen, en 2) te gebruiken met univariate statistieken, zoals gemiddelde van de waarden [2].
Deze methode is gemakkelijk te implementeren. Voordat een attribuut/variabele wordt verwijderd moet er eerst gekeken worden naar de relevantie van de variabele ten opzichte van de data. Jammer genoeg moeten belangrijke variabelen bewaard worden, ook al heeft deze veel ontbrekende waarden. Case deletion is toepasbaar als er maar een klein gedeelte missing is. Voor andere gevallen waar de steekproef omvang niet groot genoeg is of waar het aantal missing erg groot is, is deze methode niet toepasbaar. Dit komt omdat case deletion meer biased schattingen laat zien dan alternatieve methodes. Case deletion mag alleen gebruikt worden in gevallen waar de ontbrekende data voldoen aan de eis van MCAR.

4.3.1 Toepassing case deletion bij EIM

Voor case deletion is het dus heel belangrijk dat er aan de MCAR conditie is voldaan. Stel dat onze data aan die conditie voldoet en case deletion wordt toegepast dan kunnen de volgende conclusies worden getrokken:

1) We willen juist wel alle gegevens hebben, het verwijderen van de missing waarden heeft voor ons helemaal geen nut. De bedoeling is juist de gehele dataset te gebruiken voor verder onderzoek. Het verwijderen van waarden is dus tegenstrijdig met wat we willen bereiken.
2) Aantal ontbrekende waarden kan wel oplopen tot 50%. Ook al geldt 1) niet, de dataset die wij krijgen bevat missing waarden die wel tot 50% kunnen oplopen. Voor case deletion is het ook belangrijk dat er een niet al te groot percentage waarden ontbrekend zijn.
3) De overgebleven data zijn niet representatief genoeg voor de hele data. Bij het toepassen van deze techniek wordt er van uitgegaan dat de overgebleven data representatief genoeg zijn om over de gehele data (aanwezig + missing) iets te zeggen. Ook al gelden 1) en 2) niet, het verwijderen van waarden kan in ons geval een vertekend beeld opleveren (voor zover dat kan). Het domein van de waarden verschilt zo veel van elkaar dat als men deze dataset zou gebruiken voor verder onderzoek dat deze niet representatief genoeg is voor de gehele data.

Case deletion kan en mag in ons geval niet gebruikt worden. Het gebruik zou volkomen nutteloos zijn om de belangrijkste reden dat we juist *wel* de ontbrekende waarden willen hebben.

4.4 Expectation Maximization (EM)

Het EM algoritme is een algemeen iteratief algoritme voor maximum likelihood[6] schattingen van incomplete data problemen. Het EM algoritme geeft vorm aan een relatief oud ad-hoc idee om missing data te behandelen, namelijk: 1) vervang de missing waarden door geschatte waarden, 2) schat de parameters, 3) schat opnieuw de missing waarden onder de aanname dat de nieuwe parameter schatting correct is, 4) schat opnieuw de parameters en de iteratie wordt weer herhaald totdat de parameters convergeren [11].
Als de ontbrekende waarden bekend zijn, dan is het schatten van de parameters rechttoe rechtaan. Hetzelfde geldt als de parameters van het model bekend zijn, dan zou het mogelijk zijn om de unbiased voorspellingen voor de missing waarden te verkrijgen. Wat is de intuïtie achter EM? Voor elke stap wordt verondersteld dat de andere stap is opgelost. Als de toekenning van de datapunten bekend is, dan kunnen de parameters worden geschat. Als de parameters van de verdeling bekend zijn, dan kan elk punt toegewezen worden aan een model. Voor deze methode is de MAR vereist.

EM methode:
-Init: Kies waarden voor de parameters
-Iteratief proces totdat parameters convergeren. De E stap vindt de conditionele verwachting van de missing data gegeven de geobserveerde data en huidige geschatte parameters, en substitueert deze verwachtingswaarde voor de missing data.
-Maximalisatie stap: voer een maximum likelihood schatting van de parameter uit, doe dit op een zodanig manier alsof er geen ontbrekende data zijn.

4.4.1 Toepassing Expectation Maximization bij EIM

Voor elk missing data patroon wordt er een model gemaakt dat de verwachte waarde y berekent. De maximalisatie stap berekent de parameter van de eerste stap. Er wordt dan begin waarden ingevuld voor de parameters. Op een gegeven moment convergeren deze parameters, en kunnen de waarden voor de missing uitgerekend worden op basis van dit model.

Er wordt hier dus aangenomen dat de verschillende missing data patroon met elkaar vergelijkbaar zijn. Wat natuurlijk niet kan, aangezien de verschillende bedrijfsgroepen, dan wel grootteklasse op een vergelijkbaar niveau wordt gezet. Er wordt dan in feite gezegd dat loodgieters dezelfde ‘prestatie leveren’ als bouwvakkers.

4.5 Algemene conclusie

De technieken die beschreven zijn kunnen theoretisch gezien gebruikt worden om de missing waarden te berekenen. Sommigen zijn zelfs in staat om de missing waarde goed te schatten, de multiple imputation techniek bijvoorbeeld. Volgens de theorie wordt er hier rekening gehouden met het gemiddelde en de standaardfouten. Jammer genoeg

[6] Een likelihood functie is een conditionele kansfunctie. Zo een functie kan er dan als volgt uitzien $L(\theta|Y)$. De likelihood is dus een functie van de parameter θ voor een vaste Y. De geïnteresseerde lezer wordt verwezen naar de literatuur.

kunnen we deze techniek niet gebruiken, zo ook de expectation maximization en case deletion techniek.
Er wordt dan aangenomen dat de data vergelijkbaar zijn met elkaar als deze technieken worden toegepast. Voor deze technieken wordt er aangenomen dat de data voldoen aan MAR of MCAR, anders kunnen deze technieken niet toegepast worden. In de praktijk is het moeilijk te bepalen of een dataset aan één van deze twee eisen voldoet. In ons geval maakt het niet uit aan welke eisen de dataset voldoet. De waarden die we hebben zijn namelijk allemaal uniek. Uniek in de zin dat we voor de groepen met de bijbehorende subgroepen maar één getal hebben. We kunnen de waarden dus niet met elkaar vergelijken of samenvoegen. De belangrijkste reden dat we deze technieken niet kunnen gebruiken is dat wij slechts over de secundaire data beschikken, de primaire data die nodig is voor het uitvoeren van deze technieken, zijn niet tot onze beschikking. Wat wij tot onze beschikking hebben zijn de totalen, op welke wijze deze totalen zijn opgebouwd is niet bekend bij ons. Van deze technieken zijn er geen bruikbare elementen gevonden.

5 Mogelijke oplossingen

In dit hoofdstuk zal een aantal mogelijke praktische oplossingen worden toegelicht. In paragraaf 5.1 en 5.2 worden twee technieken beschreven die afkomstig zijn uit de literatuur. In paragraaf 5.3 en 5.4 zijn twee methodes beschreven die voor dit specifieke probleem zijn bedacht. Deze hebben geen theoretische ondersteuning, in die zin dat ze niet afkomstig zijn uit de theorie, maar eerder een 'intelligente' oplossing zijn van het probleem.

5.1 Verwachtingswaarde uitrekenen

In een matrix is het mogelijk om het binnenwerk uit te rekenen mits de randtotalen bekend zijn. Onder de aanname dat de randtotalen en het totaal bekend is kan gebruik worden gemaakt van de volgende formule om de verwachte waarden van het binnenwerk te berekenen:

$$X_{ij} = \frac{\sum_{i=1}^{n} X_{ij} * \sum_{j=1}^{m} X_{ij}}{\sum_{i=1}^{n} \sum_{j=1}^{m} X_{ij}}$$

De verwachte waarde voor een cel wordt dus berekend door het subtotaal in die kolom te vermenigvuldigen met het subtotaal van die rij, gedeeld door het totaal van alle cellen. Dit wordt voor alle cellen gedaan. Zo ontstaat er een dataset met alleen maar berekende waarden. Deze methode wordt o.a. bij de chi-kwadraat toets gebruikt voor het uitrekenen van de chi-kwadraat waarde, de twee gebruikte (klasse)variabelen zijn namelijk onafhankelijk van elkaar onder de chi-kwadraat toets.

5.1.1 Toepassing verwachtingswaarde berekenen bij EIM

De chi-kwadraat toets zal hier dus niet worden gebruikt, aangezien we geen statistische toets willen uitvoeren. Alleen de formule om de verwachte waarde in een cel uit te rekenen wordt gebruikt. Bij het gebruik van deze methode wordt dus onafhankelijkheid verondersteld. Aangezien er niet meer gegevens over de data beschikbaar zijn, is het aannemen van onafhankelijkheid acceptabel.
In Tabel 5-1 is te zien wat er minimaal aan gegevens beschikbaar moet zijn voor het berekenen van de ontbrekende waarden.

BGR/GKL	1	10	11	13
512	.	.	.	10
513	.	.	.	15
514	.	.	.	20
490	16	5	24	45

Tabel 5-1 één statistic met alleen de subtotalen bekend

In deze tabel zijn waarden te zien van één statistic verdeeld over de klassenvariabelen. Met in de laatste rij de totalen van de GKL's en in de laatste kolom de totalen van de

BGR's en '.' ontbrekend. Dus de '10' stelt het totaal voor van BGR 512 en de '5' het totaal van GKL 10. De ontbrekende waarden worden als volgt uitgerekend:

512 / 1 : (16x10) / 45 = 3.56
512 / 10 : (5x10) / 45 = 1.11
512 / 11 : (24x10) / 45 = 5.33
513 / 1 : (16x15) / 45 = 5.33
513 / 10 : (5x15) / 45 = 1.67
513 / 11 : (24x15) / 45 = 8
514 / 1 : (16x20) / 45 = 7.11
514 / 10 : (5x20) / 45 = 2.22
514 / 11 : (24x20) / 45 = 10.67

De nieuwe ingevulde tabel staat in Tabel 5-2:

BGR/GKL	1	10	11	13/totaal
512	3.56	1.11	5.33	10
513	5.33	1.67	8	15
514	7.11	2.22	10.67	20
490/totaal	16	5	24	45

Tabel 5-2 subtotalen met ingevulde waarden van één statistic

Hier worden de ontbrekende waarden van één statistic uitgerekend. Op dezelfde wijze kan ook de andere statistics worden uitgerekend. De ontbrekende waarden kunnen dan met behulp van deze formule uitgerekend worden, onder de voorwaarde dat de randtotalen bekend zijn.
Deze methode is niet alleen in staat om de missing waarden te berekenen, maar de berekende waarden van de rijen en kolommen sommeren ook tot het totaal. Een punt waar rekening dient te worden gehouden, is dat er ook negatieve waarden in de randtotalen kunnen voorkomen. Als de randtotalen positief zijn, dan zijn de berekende waarden ook positief. Als er negatieve waarden in de randtotalen voorkomen dan kunnen de berekende waarden afhankelijk van het eindtotaal positief dan wel negatief zijn, gewenst of ongewenst.

5.2 Genetische Algoritme (GA)

De GA is een stochastische globale zoekmethode die de metaforen van natuurlijke biologische evolutie imiteert. GAs werkt op een populatie van potentiële oplossingen waar het principe van 'survival of the fittest' wordt toegepast, in de hoop om beter en betere schattingen te maken van de oplossing. Bij elke generatie wordt er een nieuwe set van schattingen gecreëerd door het proces van het selecteren van individuen, deze voldoen aan de restricties. Deze worden vervolgens samengevoegd door gebruik te maken van operatoren van natuurlijke genetics. Dit proces leidt tot een evolutie van populaties van individuen die beter passen in hun omgeving dan de individuen van waaruit ze gecreëerd zijn, net als in de echte natuurlijke toepassing [4].

In Figuur 5-1 staat in pseudo-code een simpele GA. Hierin staan de basis componenten die nodig zijn voor het uitvoeren van GA [7]. De populatie op tijdstip t wordt voorgesteld door een tijdsafhankelijke variabele P, met begin populatie van random schattingen op P(0).

```
Procedure GA
Begin
        t=0;
        initialiseer P(t);
        evalueer P(t);
        herhaal totdat aan conditie is voldaan
        begin
                t=t+1;
                selecteer P(t) van P(t-1);
                reproduceer paren in P(t);
                evalueer P(t);
        eind
eind
```

Figuur 5-1 pseudo code van een simpel Genetic Algorithm

GA kan worden gebruikt voor veel toepassingen zoals: roosterprobleem, locatieprobleem, investeringsprobleem, transportprobleem en andere problemen.

5.2.1 Toepassing genetische algoritme bij EIM

Om te bekijken wat de toepassingen zijn van zo een GA algoritme is de software Evolver gedownload (http://www.palisade-europe.com/html/evolver.html), dit is een add-on voor Excel. Met deze solver kan een 'fitness' functie, aanpasbare cellen en restricties worden opgeven. Voor dit doel is er één variabele bekeken. In Tabel 5-3 is een vereenvoudigde vorm van de gebruikte tabel te zien.

	A	B	C	D	E	F
1		11	12	1	15	subtotaal
2	512	0	0	0	0	9209
3	513	0	0	0	0	1029
4	1747	0	0	0	0	48284
5	1748	0	0	0	0	18283
6	1749	0	0	0	0	3855
7	1750	0	0	0	0	3512
8	subtotaal	42626,76	35081,45	4711,586	1752,20	84172

Tabel 5-3 vereenvoudigt tabel van 1 variabele

De cellen die kunnen worden aangepast, zijn B2:E7. Als fitness functie is de cel F8 opgegeven die zo dicht mogelijk bij 84172 moet komen en waarin staat dat dit gelijk is aan de som van F2:F7, som van B8:E8, en een straf als de velden die aangepast kunnen boven de subtotalen van de desbetreffende rij/kolom komen. Na meer dan een uur het

programma te hebben gedraaid vindt Evolver een waarde voor de fitness functie die ongeveer 100 eenheden verschilt van de opgegeven waarde.
Er is ook geprobeerd om meer restricties op te geven voor Evolver, maar dan wordt er geen oplossing gevonden. Het programma lijkt dan 'verdwaald' te zijn in een oneindige loop, waar het programma maar geen geschikte oplossing kan vinden.

Voor kleinschalig probleem lijkt Evolver zijn werk goed te doen. Maar als het probleem groter en dus complexer wordt, lijkt Evolver daar problemen mee te hebben. Op dit moment is het niet bekend waar het probleem aan ligt, of er is een onjuiste functie en geen geschikte startwaarden opgegeven, of het programma is zodanig geprogrammeerd dat het niet overweg kan met meerdere restricties, of het genetische algoritme is niet geschikt voor het probleem.

Theoretisch gezien kan het genetische algoritme een oplossing vinden, volgens het algoritme geldt namelijk 'survival of the fittest', waar dus een 'beste' oplossing gevonden wordt.
In ons geval, worden er nullen ingevuld voor de startwaarden, er kan dan afgevraagd worden of dit dan wel geschikte startwaarden zijn. Ook het langdurig runnen van het programma is een nadeel van deze techniek.

5.3 Rekenen per cel

De volgende methode werd bedacht door een voormalige werknemer bij EIM. De filosofie achter deze methode is dat alle verantwoordelijkheid wordt gelegd bij de individuele cel. De cel is zelf verantwoordelijk om zichzelf te testen, of zichzelf te onthullen c.q. bij te schatten. Het onthullen gebeurt d.m.v. de aanwezige rekenregels. Er wordt dan gekeken of er niet met de aanwezige rekenregels een cel berekend kan worden. Om zichzelf te onthullen gaat de cel op zoek naar informatie uit zijn directe omgeving. Voor het bijschatten wordt er gebruik gemaakt van verhoudingen, er wordt dan gezocht naar de verhouding van een andere groep (met waargenomen waarde) met het totaal [5]. Hier zijn verschillende wegen mogelijk, namelijk via GKL regels, via BGR regels en andere. Jammer genoeg is deze methode niet volledig getest op allerlei aspecten. Momenteel is er geen tijd om deze verder te onderzoeken.

5.4 Middelen over de missing waarden

Missing waarde met exact één missing in een rekenkundige relatie kunnen eenvoudig uitgerekend worden. Als een missing waarde kan worden opgelost, dan wordt er gekeken of er ook niet in andere rekenkundige relaties een andere missing waarde kan worden opgelost. Deze stap stopt als er geen missing waarde meer voldoet aan de eis van exact één ontbrekend in een relatie of alle mogelijke relaties zijn al nagelopen.
Er dient opgemerkt te worden dat op deze manier van berekenen, eventuele rekenfouten/onjuistheden worden overgedragen op de overige nog in te ramen data.

Voor meer dan één missing waarde in een relatie wordt de waarde van een missing berekend door het gemiddelde van het verschil tussen de som over alle waarden in die relatie en de som over de non-missing waarden [1]. Voor m het aantal missing waarden, x_i de waarde van i-de non-missing, x_j de waarde van j-de missing en S_x de som over x_i en x_j geldt

$$x_j = \frac{S_x - \sum_i x_i}{m}$$

Zo kunnen alle missing waarden worden berekend dan wel worden afgeleid. Voor het aansturen van de subroutine die de missing waarden berekend, zijn er een aantal varianten bedacht.

1) Het aantal missings in de rekenregel precies gelijk aan één.
2) Het aantal missings in de rekenregel gelijk aan een van tevoren vastgesteld getal.
3) Het aantal missings in de rekenregel tenminste gelijk aan één maar niet groter dan een van tevoren vastgesteld getal.
4) Het aantal missings in de rekenregel tenminste gelijk aan één.

Bij de bovenstaande varianten wordt er gebruik gemaakt van het middelen, wanneer en hoe dat gebeurd wordt door de gehanteerde variant bepaald. Deze methode wordt bij de huidige onderzoeken toegepast.

5.5 Algemene conclusie

De beschreven technieken hebben allemaal hun voordelen en nadelen. Zo zijn de methoden om de verwachtingswaarde van een cel uit te rekenen en het middelen over de cellen inderdaad in staat om waarden te schatten voor de missing waarden. Een groot nadeel van deze twee methoden is dat de standaardfouten erg worden onderschat. Als er ergens in de data een uitschieter voorkomt, dan worden de geschatte data nogal vervormd. Een voordeel van de verwachtingswaarde methode t.o.v. het middelen is dat de verwachtingswaarde methode rekening houdt met meerdere dimensie, terwijl het middelen maar over één dimensie gaat. Een ander voordeel is dat voor het berekenen van de verwachtingswaarde de onafhankelijkheid tussen de klassenvariabele wordt verondersteld.
Voor het rekenen per cel is (nog) niet precies uitgezocht hoe dat in de praktijk werkt. Maar een nadeel bij het nemen van verhoudingen voor het uitrekenen van onbekende waarden is dat er eigenlijk wordt verondersteld dat de groepen onderling vergelijkbaar zijn met elkaar. Hierdoor ontstaan er resultaten die eigenlijk niet gewenst zijn.
Het genetische algoritme kan theoretisch gezien de missing waarden schatten. Met ‘simpele’ problemen is het genetische algoritme goed in staat deze op te lossen. De vraag is of dit algoritme ook overweg kan met problemen die veel restricties bevatten. Als die niet wordt opgeven, dan bestaat er een kans dat het algoritme een lokaal maximum vindt. Er wordt dan naar een makkelijke oplossing gezocht, in ons geval is dat er dan veel nullen worden ingevuld, maar zo dat de som van een rij/kolom nog wel klopt. Hoewel de data dan wel consistent zijn, is dit niet wat we willen bereiken. Ook speelt de lange looptijd om een oplossing te vinden een rol.

Hoewel de beschreven methoden allemaal in staat zijn om ‘redelijke’ waarden te vinden voor een ontbrekende waarde, kleven er aan al die methoden nadelen. Het is belangrijk om een methode te kiezen die het minst ‘erg’ is. De methode van de verwachtingswaarde uitrekenen lijkt vergeleken met de andere methoden, het meeste voordeel op te leveren.

6 Algemene aanbeveling/conclusie

6.1 Aanbeveling naar aanleiding van de dataset

Het startwaarde probleem ziet er op het eerste gezicht eenvoudig uit. Namelijk het vinden van betere startwaarden dan de nullen die er nu zijn ingevuld. Maar schijn bedriegt, want bij nader onderzoek blijkt dat dit probleem veel complexer is. Dit komt doordat de dataset die gebruikt wordt erg veel variabelen bevat. Ook hoort er bij elke dataset een aantal rekenregels waar de data aan moeten voldoen. Hoewel de betekenissen van de variabelen niet van belang is bij het oplossen van het probleem, is het toch aangeraden om hier meer van te weten. Dit geeft namelijk meer gevoel over de data, zodat de gebruikte procedures beter begrijpbaar zullen zijn. Het is daarom aan te raden om de rekenregels van paragraaf 3.2 en Bijlage 1, eventueel met Figuur 2-6 en/of Tabel 2-1 door te nemen. Bij dit probleem draait het om de onthulfase. Wat er tijdens de NLP-fase gebeurt wordt niet als het ware probleem beschouwd.

Een aantal feiten op een rij die het oplossen van het probleem bemoeilijken:

1) De dataset bevat allerlei fouten.
2) Een variabele kan op meer dan één manier worden uitgerekend.
3) De BGR- en de GKL- rekenregels kunnen alleen de ‘+’ operator bevatten. De VAR-rekenregels kunnen alleen de ‘+’ en de ‘-‘ operator bevatten.

Bij feit nummer 1 is de volgorde van het toepassen van de rekenregels belangrijk. Er zijn namelijk variabelen die fouten bevatten. Als deze variabelen als één van de eersten worden gebruikt voor het onthullen van andere variabelen, dan worden de fouten die deze variabelen bevatten doorgegeven aan de andere variabelen die in de desbetreffende rekenregel worden gebruikt. Als er in die rekenregel ook nog andere ‘foute’ variabelen worden gebruikt, dan word de fout alleen maar erger. Dit kan dan voor de hele dataset doorwerken. Welke rekenregel eerst toepassen en welke daarna is dus belangrijk voor de betrouwbaarheid van de gegevens.

Hoewel feit nummer 2 het oplossen van het probleem bemoeilijkt, heeft dit feit ook een voordeel. Namelijk als een variabele ontbrekend is en deze kan niet met de ene rekenregel worden onthuld, dan kan dit misschien wel met een andere rekenregel (waar dezelfde variabele ook in voorkomt). Op deze manier worden sommige variabelen dus vaker gebruikt dan anderen. De vaak gebruikte variabelen komen dan in zoveel rekenregels voor, dat ze als ‘belangrijk’ gezien kunnen worden. Het veranderen van zo een variabele heeft dan invloed op vele andere. Aangezien deze variabelen als belangrijk gemerkt worden is het ook logisch om ‘iets’ met deze variabelen te doen. Bijvoorbeeld de rekenregels die deze variabelen bevatten als eerste toe te passen, maar rekeninghoudend met feit nummer 1 is dit waarschijnlijk niet verstandig.
Voor variabelen die in veel rekenregels voorkomen, is de oplossingsruimte ook kleiner dan voor variabelen waar niet het geval is. Aangezien de rekenregels ook als restricties kunnen worden gezien. Door de vele restricties die wordt opgelegd voor een variabele wordt de oplossingruimte dus ook kleiner.

In feit nummer 3 staat dat de rekenregels allemaal lineair zijn. Door de verschillende betekenissen van de variabelen kunnen de VAR-rekenregels ook de '-' operator bevatten. Deze '-' operatoren kunnen voor enorme problemen veroorzaken. Dit is het geval als er meer dan één variabele in een dergelijke rekenregel onbekend is. Bijvoorbeeld: A = - B + C, als alleen A bekend is, dan kunnen B en C door het min-teken voor de B alle waarden aannemen, ongeacht wat de waarde van A is. Dit kan worden verholpen als B of C ook met behulp van andere rekenregels kan worden uitgerekend, zodat de oplossingsruimte voor die variabelen kleiner worden. Aangezien er een aantal vaste rekenregels zijn, is het belangrijk dat de rekenregels op een slimme manier wordt toegepast (het is niet mogelijk rekenregels toe te voegen dan wel te verwijderen wanneer het uitkomt). Een oplossing is het eerst onthullen van de BGR*GKL dimensie en daarna pas de VAR dimensie. Dit omdat de BGR en GKL rekenregels geen negatieve operatoren kunnen hebben. Dit wordt gebruikt in de hoop dat B en/of C al in een vroeg stadium wordt uitgerekend, zodat deze in het latere stadium (VAR regels) geen problemen kunnen veroorzaken.

6.2 Aanbeveling wat betreft de oplossingen

De technieken die in de literatuur beschreven staan, zijn niet direct toepasbaar. Al deze technieken veronderstellen namelijk dat de primaire data beschikbaar zijn, wij beschikken alleen maar over de secundaire data. Ook zijn er binnen deze technieken geen elementen gevonden die bruikbaar zijn voor het probleem.
Er zijn wel een aantal ideeën bedacht die mogelijk een oplossing kunnen bieden aan het probleem, maar de meeste ideeën zijn niet uitgewerkt. Deze ideeën houden geen tot weinig rekening met de bovenstaande feiten. Ik denk dat het startwaarde probleem betere/snellere oplossing geeft als bovenstaande feiten eerst wordt aangepakt dan wel wordt opgelost. Hoewel er aan deze feiten niks gedaan kan worden, kan er wel oplossing worden bedacht, zodat de problemen die deze feiten meebrengen worden opgevangen of in ieder geval worden verminderd.

Referentie

[1] Berkel van, P. (2004). Verslag van bevindingen: Systeemaanpassing BLISS Inkomens, (interne verslag van EIM)

[2] Chantal, Kim and Suchindran C. Multiple imputation for missing data. (MI)

[3] Feelders, A.J. An overview of model based reject inference for credit scoring (2-5) (MAR)

[4] Eiben, A.E. and Smith, J.E. (2003). Introduction to evolutionary computing, (15-24) (GA)

[5] Hennen, W. (2004). Volledig geautomatiseerd dataverwerkingsproces BLISS lonen: Mogelijk en Wenselijk?, (interne verslag van EIM)

[6] Little, Roderick J.A and Rubin, Donald B. (1987). Statistical analysis with missing data, Wiley series in probability and mathematical statistics.

[7] MATLAB genetic algorithm toolbox user's guide v1.2 (1-3 - 1-10) (GA)

[8] SAS manual Ch 09 – The MI procedure V8.02 (133-149) (MI)

[9] Schafer, Joseph L and Olsen Maren K. (1998). Multiple imputation for multivariate missing-data problems: a data analyst's perspective, The Pennsylvania State University (MI)

[10] Wayman, Jeffrey C. (2003). Multiple Imputation For Missing Data: What Is It And How Can I Use It? , Johns Hopkins University (MI)

[11] Weiss, Yair. Motion segmentation using EM – a short tutorial, Cambridge (EM)

Bijlage 1

Betekenis van de variabelen

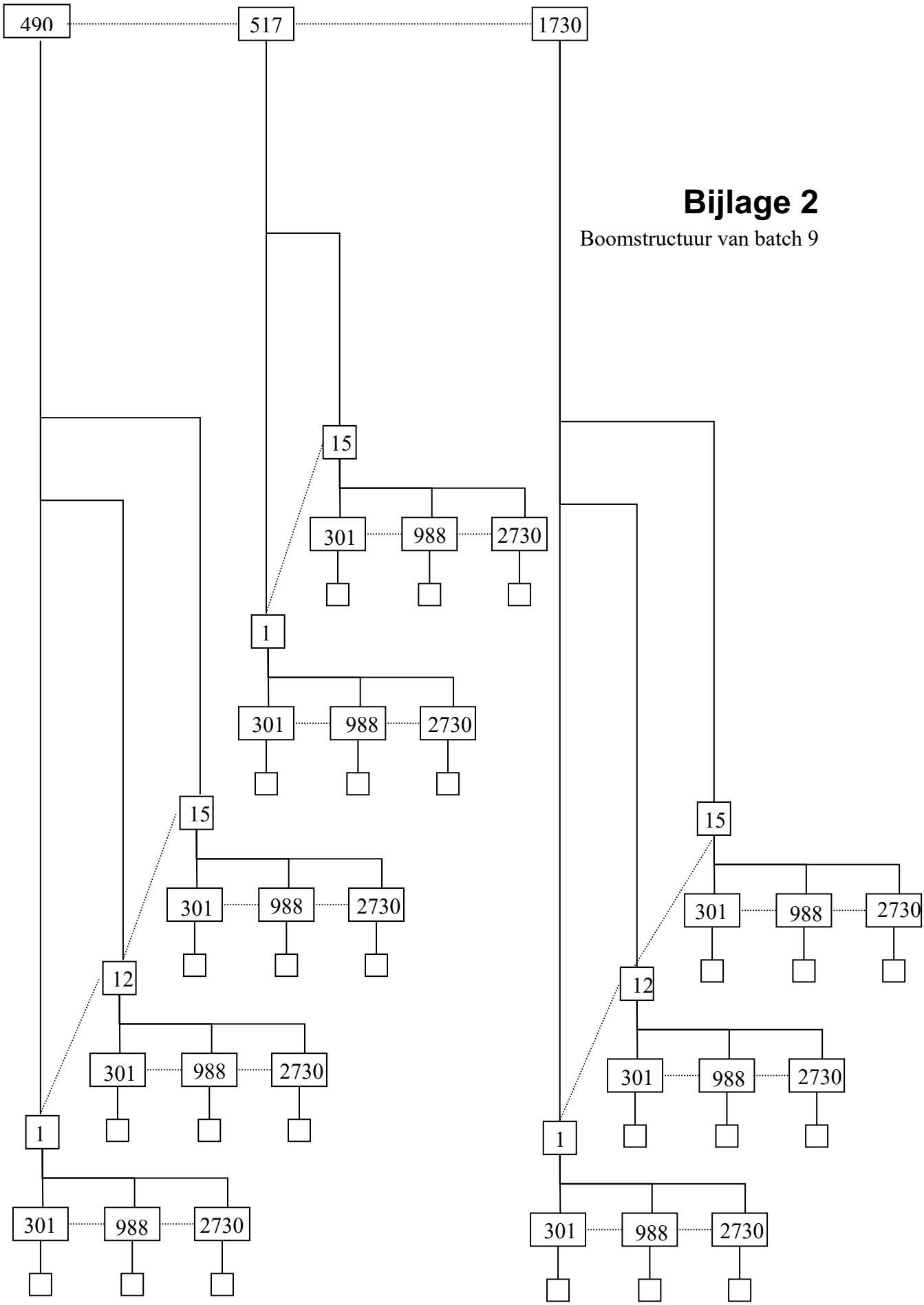
index	BGR	BGR_omschrijving
1	490	Bouwnijverheid
2	512	Elektrotechnische bouwinstallatie
3	513	Isolatiewerkzaamheden
4	514	Loodgieters-, fitterswerk; installatie van sanitair, centrale verwarmings- en luchtbehandelingsapparaten
5	515	Loodgieters-, fitterswerk; installatie van sanitair
6	516	Installatie van centrale verwarmings- en luchtbehandelingsapparaten
7	517	Overige bouwinstallatie
8	521	Afwerken van vloeren en wanden
9	522	Schilderen en glaszetten
10	524	Verhuur van bouw- en sloopmachines met bedienend personeel
11	1747	Burgerlijke- en utiliteitsbouw
12	1748	Grond-, water- en wegenbouwbedrijven
13	1749	Heien en andere funderingswerkzaamheden; vlechten van betonstaal; overieg gespecialiseerde werkzaamheden in de bouw n.e.g.
14	1750	Stukadoren; timmeren; overige afwerking van gebouwen

index	GKL	GKL omschrijving
1	1	0 werknemers
2	10	0 t/m 9 werknemers
3	11	10 t/m 99 werknemers
4	12	100 en meer werknemers
5	13	Totaal
6	14	10 en meer werknemers
7	15	1 t/m 9 werknemers

index	VAR	Omschrijving
22	302	Saldo voorzieningen
32	814	Onttrekkingen aan voorzieningen
35	988	Toevoegingen aan voorzieningen
39	1222	Beginvoorraad grond- en hulpstoffen incl. verpakkingsmiddelen
41	1225	Eindvoorraad grond- en hulpstoffen incl. verpakkingsmiddelen
46	1314	Voorraadmutaties grond- en hulpstoffen
64	2727	Omzet nettoverhuur Nederland
65	2729	Omzet nettoverhuur buitenland
66	2730	Omzet nettoverhuur

Bijlage 2

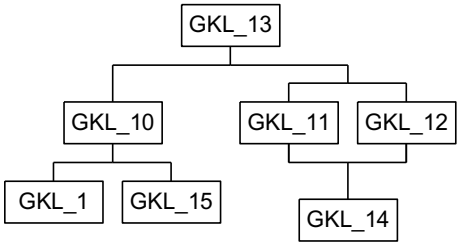
Boomstructuur van batch 9



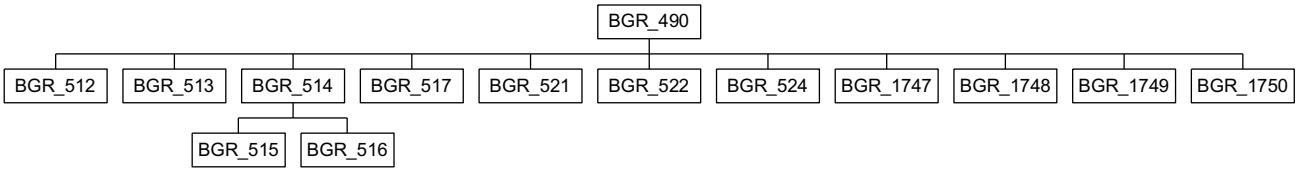
Bijlage 3

Hier wordt de hiërarchie geven de groepen van batch 9. Voor de groepen BGR en GKL is er altijd maar één boom, aangezien deze tot een totaal (bovenste knoop) optellen. Maar voor de VAR groep is dit niet het geval, deze kan uit meerdere bomen bestaan.

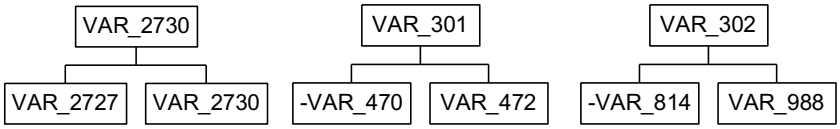
Hiërarchie van GKL



Hiërarchie van BGR

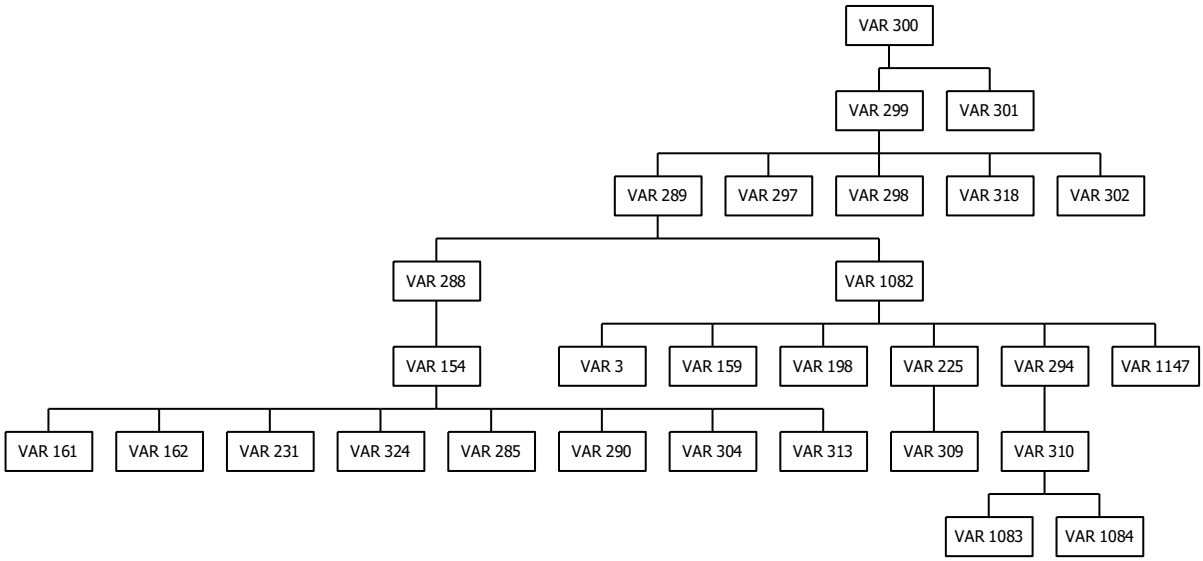


Hiërarchie van VAR



Onderstaande boom is afkomstig van een andere batch, hieruit is te zien dat een boom erg veel vertakkingen kan hebben.

VARIABELEN-HIËRARCHIE



Bijlage 4

Een voorbeeld van hoe de ontbrekende waarden kunnen worden onthuld.

Laat het volgende bekend zijn van een bepaalde VAR:

BGR \ GKL	1	15	10
515	2	.	.
516	.	2	.
514	3	.	10

Waarbij de volgende rekenregels gelden:

- BGR_514 = BGR_515 + BGR_516
- GKL_10 = GKL_1 + GKL_15

Met de beschikbare gegevens en de rekenregels is dit voldoende om alle ontbrekende waarden af te leiden. Er wordt nu in elke stap telkens de variabele afgeleid waar in de rekenregel maar één ontbrekend is, deze is dik en cursief gedrukt.

stap 1:

BGR \ GKL	1	15	10
515	2	.	.
516	***1***	2	.
514	3	*7*	10

stap 2:

BGR \ GKL	1	15	10
515	2	***5***	.
516	1	2	***3***
514	3	7	10

stap 3:

BGR \ GKL	1	15	10
515	2	5	*7*
516	1	2	3
514	3	7	10

Het afleiden moet dus stap voor stap gebeuren. In stap 1 wordt de indruk gewekt dat er tegelijk 2 waarden worden afgeleid. Dus is dus niet het geval, er wordt eerst de ‘1’ afgeleid en daarna de ‘7’, of andersom. Dit hangt af welke rekenregel eerst wordt toegepast. In dit geval wordt dus eerst de BGR rekenregels toegepast en daarna de GKL rekenregel.