Aantal clusters in een dataset vinden met interne prestatiemaatstaven

Auke Tas, 400701

supervisor: Dr. M. van de Velden

Abstract

In dit paper worden verschillende methoden vergeleken om tegelijk datasets te clusteren en het aantal dimensies terug te brengen. De methoden worden vergeleken via een simulatiestudie waarin de allocaties voor verschillende soorten datasets worden bekeken. Vervolgens worden deze allocaties met behulp van verschillende prestatiemaatstaven met elkaar vergeleken. Uiteindelijk blijkt dat de specifieke eigenschappen van de dataset grote invloed hebben op de uitkomsten van de prestatiemaatstaven en dat over het algemeen de silhouette score en de Krzanowski-Lai index het best het juiste aantal clusters herkennen.

1. Introductie

Clustering is een populaire uitdaging in machine learning en data analyse. Het betreft het opdelen van een dataset in verschillende groepen waarbij de datapunten binnen een groep (cluster) op elkaar lijken en de datapunten in verschillende groepen juist niet op elkaar lijken. Om dit probleem computationeel te vergemakkelijken wordt meestal eerst het aantal dimensies teruggebracht waarin datapunten bekeken worden (Keogh en Mueen, 2010). Er zijn verschillende manieren om dit te doen afhankelijk van het soort data dat gebruikt wordt. Een probleem is dat bekende dimensie-reductie technieken niet per se een gereduceerde ruimte opleveren waarin de clusters beter te onderscheiden zijn. Dit komt omdat dimensie-reductie technieken zoveel mogelijk variantie van de data proberen weer te geven in zo min mogelijk dimensies. Dit betekent dat als er variabelen zijn die veel variantie bevatten maar niet bijdragen aan de structuur van de clusters de dimensie-reductie technieken hun best zullen doen om deze variantie te bevatten in de lagere dimensies, soms ten koste van variabelen die wel bijdragen aan de cluster structuur (Vichi en Kiers, 2001). Om dit probleem tegen te gaan zijn verschillende methoden bedacht die als doel hebben om tegelijkertijd een goede allocatie van de data in clusters te maken en een gereduceerde ruimte te vinden die de verschillen tussen de gevonden clusters zo goed mogelijk weergeeft (Van de Velden et al., 2016).

Een van de meest problematische aspecten aan clustering die door deze methoden niet wordt opgelost is het bepalen van het aantal clusters in een dataset (Cordeiro de Amorim, 2016). Aangezien de onderliggende verdeling van de data in de praktijk vaak onbekend is is het aantal aanwezige clusters niet triviaal en hangt dit ook af van de interpretatie van de oplossing. Er zijn door de jaren heen verschillende statistieken bedacht om de allocatie van observaties aan clusters te beoordelen. Het vergelijken van deze statistieken zou een leidraad kunnen zijn bij het kiezen van het aantal clusters. In dit paper worden een paar ervan, te weten Silhouette score, Calinski-Harabasz index, Krzanowski-Lai index en een door Dolnicar en Leisch voorgestelde methode die gebruikt maakt van bootstrapping in een simulatie studie met elkaar vergeleken om te beoordelen of de waarde van deze statistieken iets kan zeggen over de juistheid van het gekozen aantal clusters.

2. Methoden

In dit paper worden verschillende methoden vergeleken om een dataset te clusteren. Alle methoden staan beschreven in Van de Velden et al. (2016) en worden hieronder kort uitgelegd. Alle methoden hebben als doel om tegelijkertijd data te clusteren en het aantal dimensies terug te brengen op een manier die zoveel mogelijk onderscheid maakt tussen de gevonden clusters. Dit om te voorkomen dat, zoals beschreven in de inleiding, de dimensiereductie dimensies oplevert die weliswaar veel variantie in de data verklaren maar die eigenlijk niet verder bijdragen aan de onderliggende structuur van de clusters. Alle onderstaande methodes kunnen gebruikt worden om categorische data te clusteren. Een categorische variabele q met q_j verschillende opties kan beschreven worden in een $n \times q_j$ indicator matrix $\mathbf{Z}$ waarbij $\mathbf{Z}(i,j)$ 1 is als observatie i in categorie j valt en 0 anders. n is in dit geval het aantal observaties. Meerdere categorische variabelen $q_1, ..., q_p$ met bijbehorende indicator matrices $\mathbf{Z}_1, ..., \mathbf{Z}_p$ kunnen weergegeven worden in een superindicator matrix $\mathbf{Z} = [\mathbf{Z}_1, ..., \mathbf{Z}_p]$. Bij welk cluster een observatie hoort kan ook als categorische variabele worden gezien en dus worden weergegeven in een indicator matrix, hierna $\mathbf{Z}_k$ genoemd.

Om data te clusteren gebruiken de methoden het k-means algoritme. K-means begint met het willekeurig kiezen van de centrums van de clusters. Meestal worden deze centrums willekeurig gekozen, maar ze kunnen ook van tevoren bepaald worden. Vervolgens worden om en om de volgende twee stappen uitgevoerd:

1. Alle observaties worden toegewezen aan het cluster met het centrum dat het dichtst bij de observatie ligt.
2. Het nieuwe clustercentrum wordt berekend als het centrum van alle punten die aan het cluster zijn toegewezen.

K-means is een iteratief algoritme dat doorgaat totdat opeenvolgende iteraties geen andere allocatie van de clusters meer opleveren. K-means werkt met een

van tevoren vastgesteld aantal clusters. Deze probeert het algoritme vervolgens zo goed mogelijk te onderscheiden. Bij een verkeerde keuze voor het aantal clusters (k) kan het zijn dat k-means geen goede allocatie oplevert.

Voor dimensie reductie van categorische data gebruiken de in dit paper gebruikte methoden correspondentie analyse. Correspondentie analyse is een manier om via een singuliere waarde ontbinding een (super)indicatormatrix uit te drukken in minder dimensies die zoveel mogelijk variantie van de indicatormatrix verklaren. Singuliere waarde ontbinding wil zeggen dat elke matrix $\mathbf{A}$ met afmeting $i \times j$ te schrijven is als $\mathbf{U}\mathbf{D}_a\mathbf{V}^\intercal$ waarbij $\mathbf{D}_a$ een $k \times k$ diagonaalmatrix is met positieve waarden $a_1, ..., a_k$ waarbij k de rank van de matrix $\mathbf{A}$ is en $\mathbf{U}$ en $\mathbf{V}$ een orthonormale basis zijn voor respectievelijk de kolommen en rijen van de matrix $\mathbf{A}$ (Greenacre, 1984). Een l-dimensionele benadering van de kolommen van $\mathbf{A}$ kan vervolgens gevonden worden door de eerste l kolommen van $\mathbf{V}$ te nemen als ze geordend zijn naar de grootte van het corresponderende diagonaalelement van $\mathbf{D}_a$. Op dezelfde manier leveren de eerste l rijen van $\mathbf{U}$ een l-dimensionele benadering van de rijen van $\mathbf{A}$. Op deze manier wordt fractie $\frac{\sum_{i=1}^{l} a_i^2}{\sum_{i=1}^{k} a_i^2}$ van de variantie verklaard.

2.1. Cluster Correspondence Analysis

Cluster correspondence analysis (hierna CCA genoemd) is een methode ontwikkeld door Van de Velden et al.(2016) om tegelijk data te clusteren en dimensie reductie toe te passen die zo goed mogelijk onderscheid maakt tussen de gevonden clusters. In plaats van naar de originele superindicatormatrix $\mathbf{Z}$ kijkt CCA naar de matrix $\mathbf{Z}_k'\mathbf{Z}$ die als rijen de k verschillende clusters heeft en als kolommen de categorische variabelen net als bij de superindicatormatrix $\mathbf{Z}$. De clusterindicatiematrix $\mathbf{Z}_k$ is in het begin nog onbekend en dient door het algoritme zo goed mogelijk bepaald te worden. In de eerste stap wordt willekeurig aan elke observatie een cluster toegekend. In volgende iteraties zorgt het algoritme voor een allocatie die zo onderscheidend mogelijke clusters oplevert. In de matrix $\mathbf{Z}_k'\mathbf{Z}$ is per cluster weergegeven hoeveel observaties aan een bepaalde waarde voldoen voor elke categorische variabele. Vervolgens wordt de matrix $\mathbf{P} = \frac{1}{np}\ \mathbf{Z}$ bepaald met hierin de proporties waarin elke waarde voor de categorische variabele voorkomt per cluster. Op deze matrix wordt vervolgens correspondentie analyse toegepast om nieuwe dimensies te vinden die de clusters zo goed mogelijk onderscheiden. In deze nieuwe ruimte wordt vervolgens het k-means algoritme toegepast om een nieuwe allocatie $\mathbf{Z}_k$ te vinden. Met deze allocatie wordt opnieuw $\mathbf{Z}_k'\mathbf{Z}$ berekend en wordt het hierboven beschreven algoritme opnieuw toegepast, net zolang totdat convergentie bereikt is en $\mathbf{Z}_k$ tussen verschillende iteraties niet meer verandert. Naast het aantal clusters moet ook het aantal dimensies tot welke de data wordt gereduceerd van tevoren bepaald worden.

2.2. MCA K-means

MCA K-means werd in 2006 voorgesteld door Hwang et al. om tegelijk data te clusteren en dimensie-reductie toe te passen. Ook deze methode gebruikt k-means en correspondentie analyse, maar geeft ze beiden een gewicht om te bepalen in hoeverre ze de oplossing beinvloeden. Dit betekent dat naast het aantal clusters en het aantal dimensies ook nog een gewicht α bepaald moet worden dat het MCA gedeelte krijgt, en hiermee automatisch een gewicht $1-\alpha$ dat het k-means gedeelte krijgt. Gegeven het gewicht α komt MCA K-means neer op het minimaliseren van de functie

$$\alpha \frac{1}{p} \sum_{j=1}^{p} \|\mathbf{Y} - \mathbf{Z}_j \mathbf{B}_j\|^2 + (1-\alpha)\|\mathbf{Y} - \mathbf{Z}_k \mathbf{G}\|^2$$

waarbij $\mathbf{Z}_j$ de jde categorische variabele is, $\mathbf{B}_j$ de gewichten die de categorische variabelen hebben gekregen in de gereduceerde ruimte, $\mathbf{Z}_k$ wederom de indicatormatrix voor cluster lidmaatschap en $\mathbf{G}$ de matrix met clustercentra. Deze functie wordt met Alternating Least Squares (de Leeuw, Young & Takane, 1976) geminimaliseerd naar $\mathbf{Y}, \mathbf{B}_j, \mathbf{G}$ en $\mathbf{Z}_k$. Dit levert het volgende algoritme op:

1. Bepaal initiele allocatie van de data $\mathbf{Z}_k$
2. Update $\mathbf{B}_j$ en $\mathbf{G}$ met $\mathbf{B}_j = (\mathbf{Z}_j'\mathbf{Z}_j)^{-1}\mathbf{Z}_j'\mathbf{Y}$ en $\mathbf{G} = (\mathbf{Z}_k'\mathbf{Z}_k)^{-1}\mathbf{Z}_k'\mathbf{Y}$
3. Bepaal een nieuwe waarde voor $\mathbf{Y}$ met behulp van de eigenwaarde vergelijking $\mathbf{Y}\mathbf{\Lambda} = (\alpha \frac{1}{p} \sum_{j=1}^{p} \mathbf{Z}_j(\mathbf{Z}_j'\mathbf{Z}_j)^{-1}\mathbf{Z}_j' + (1-\alpha)\mathbf{Z}_k(\mathbf{Z}_k'\mathbf{Z}_k)^{-1}\mathbf{Z}_k')\mathbf{Y}$ waarbij $\mathbf{\Lambda}$ een diagonaalmatrix is met als elementen op de diagonaalas de eigenwaarden van $\mathbf{Y}$
4. Bepaal een nieuwe allocatie $\mathbf{Z}_k$ door k-means toe te passen op de nieuwe $\mathbf{Y}$
5. Ga terug naar stap 2 tot convergentie is bereikt

3. Prestatiemaatstaven

Er zijn verschillende manieren om de allocatie die een clusteringalgoritme oplevert te vergelijken. Het belangrijkste onderscheid tussen deze manieren is wat in academische literatuur interne en externe prestatiemaatstaven genoemd worden. Een externe prestatiemaatstaf gebruikt meer input dan alleen de dataset en een allocatie van de dataset in clusters. Een voorbeeld hiervan is de Adjusted Rand Index (ARI) die naast de allocatie die een algoritme oplevert nog een tweede allocatie gebruikt en deze twee allocaties vergelijkt. In empirische toepassingen zijn externe prestatiemaatstaven niet altijd bruikbaar omdat de extra informatie die de prestatiemaatstaf vereist niet altijd beschikbaar is, maar in simulaties kunnen ze gebruikt worden om methoden te vergelijken (Van de Velden et al., 2016). Interne prestatiemaatstaven zijn statistieken waarbij alleen wordt gekeken naar de dataset en de allocatie die beoordeeld moet worden. Een goede allocatie moet over het algemeen aan twee zaken voldoen. Ten

eerste moeten de observaties die in hetzelfde cluster geplaatst zijn op elkaar lijken, oftewel ze moeten dicht bij elkaar liggen in de gekozen ruimte. Ten tweede moeten observaties die niet in hetzelfde cluster zijn geplaatst juist niet op elkaar lijken, en dus ver van elkaar af liggen. Op deze manier is er een duidelijk onderscheid tussen de gevormde clusters. Deze twee vereisten heten compactheid en scheiding (Liu et al., 2010). Aangezien in dit paper geen kennis verondersteld wordt buiten de dataset en de allocatie die een algoritme oplevert worden alleen interne prestatiemaatstaven bekeken.

3.1. Silhouette score

De silhouette score gebruikt om compactheid te meten de gemiddelde afstand van een punt tot alle andere punten in zijn cluster. Om scheiding te meten wordt voor een punt de gemiddelde afstand tot alle punten in elk ander cluster gemeten. Het cluster waarbij deze gemiddelde afstand het kleinst is wordt het naburige cluster genoemd en deze wordt gebruikt bij het berekenen van de silhouette score. Als $a(i)$ de gemiddelde afstand naar punten uit hetzelfde cluster is en $b(i)$ de gemiddelde afstand naar punten uit het naburige cluster dan wordt de silhouette score van een punt berekend als $\frac{b(i)-a(i)}{max(b(i),a(i))}$. De Silhouette score ligt altijd tussen -1 en 1 waarbij een negatieve score aangeeft dat een punt in het verkeerde cluster is ingedeeld (of in elk geval dat de gemiddelde afstand tot punten in een ander cluster kleiner is). Vervolgens wordt het gemiddelde van de silhouette score van alle punten genomen welke uiteraard ook tussen -1 en 1 ligt. Voor interpretatie van de Silhouette score geldt: hoe hoger de gemiddelde Silhouette score, hoe sterker de gevonden cluster structuur.

3.2. Calinski-Harabasz index

De Calinski-Harabasz index zet de variantie binnen elk cluster af tegen de variantie tussen de clusters. Om compactheid te evalueren worden per cluster van alle punten in dat cluster de kwadratische afstand tot het centrum van het cluster berekend. Vervolgens wordt dit geaggregeerd over alle clusters. Om scheiding te meten wordt de variantie tussen alle clusters gemeten. Dit gebeurt door van elk cluster het centrum te nemen en van dit punt de kwadratische afstand tot het centrum van alle data te berekenen. Deze afstand wordt vervolgens weer geaggregeerd over alle clusters. Als we de compactheid $SSRA$ noemen en de scheiding $SSRB$ is de Calinski-Harabasz te berekenen door $\frac{(N-K)SSRA}{(K-1)SSRB}$ waarbij de termen $K-1$ en $N-K$ zijn om de het gemiddelde van de afstanden tussen en binnen de clusters te nemen.

3.3. Krzanowski-Lai index

Krzanowski en Lai (1988) ontwikkelden een prestatiemaatstaf die de kwadratische afstand van observaties tot het centrum van hun cluster bekijkt en deze vergelijkt voor verschillende aantal clusters. Voor een aantal waarden van k wordt voor elke observatie de kwadratische afstand tot het centrum van het bijbehorende cluster genomen. Deze afstanden worden vervolgens over de hele dataset geaggegreerd en hierna SSR_k genoemd. Krzanowski en Lai beschrijven

vervolgens het verschil tussen een allocatie in k-1 en k clusters $DIFF_k$ als $(K-1)^{\frac{2}{J}} \times SSR_{k-1} - K^{\frac{2}{J}} \times SSR_k$, waarbij J het aantal variabelen is dat bekeken wordt. Het idee van de Krzanowski-Lai index is dat als er een juist aantal clusters k^* bestaat dat dit aantal een relatief grote verbetering in SSR_k zal opleveren, en $DIFF_k$ dus groot zal zijn voor $k = k^*$. Daarentegen zal $DIFF_k$ juist klein zijn voor $k > k^*$. Dit betekent dat de Krzanowski-Lai index gedefinieerd als $\frac{DIFF_k}{DIFF_k-1}$ maximaal zijn voor $k = k*$.

3.4. Dolnicar-Leisch bootstrap methode

Dolnicar en Leisch (2010) focussen op de stabiliteit van een allocatie. Dit wil zeggen hoe gevoelig de allocatie is voor veranderingen in de data. Ze wijzen erop dat de dataset die voorhanden is vaak slechts een steekproef is die willekeurig uit een grotere populatie is getrokken en dat de willekeur die optreedt bij het trekken van deze steekproef vaak over het hoofd wordt gezien bij het evalueren van oplossingen van algoritmen. Het is moeilijk om deze willekeur te elimineren aangezien er vaak maar een steekproef beschikbaar is van de data en er geen onderliggende verdeling bekend is. Dolnicar en Leisch lossen dit op door met behulp van bootstrapping (Efron en Tibshirani, 1993) B nieuwe steekproeven te genereren met de empirische distributie van de beschikbare dataset. Elk van deze steekproeven levert een bepaalde allocatie C_b op in een van tevoren gekozen aantal clusters. In het geval van de algoritmen die in dit paper worden vergeleken wordt met een allocatie een aantal clustercentra bedoeld waarbij elke observatie wordt toegewezen aan het centrum dat het dichtstbij de observatie ligt. Vervolgens wordt de oorspronkelijke dataset volgens deze B verschillende allocaties opnieuw opgedeeld. Het idee van Dolnicar en Leisch is dat als de clusters in de dataset duidelijk onderscheidend zijn ze ook in de gebootstrapte steekproeven teruggevonden kunnen worden. Dit zou betekenen dat de B allocaties die hierboven gevonden zijn op dezelfde manier de observaties in de oorspronkelijke dataset aan clusters toewijzen. Om dit te testen wordt het verschil tussen de verschillende allocaties bepaald. Er zijn verschillende manieren om het verschil tussen twee allocaties te bepalen. In dit geval zal de Adjusted Rand index (Hubert en Arabie, 1985) gebruikt worden. Om te zorgen dat de verschillen die gevonden worden onafhankelijk zijn van elkaar worden slechts $\frac{B}{2}$ paren bekeken, te weten C_1 en C_2, C_3 en C_4, ... , C_{B-1} en C_B. Hoe kleiner het verschil tussen de verschillende allocaties, dus hoe hoger de waarde van de Adjusted Rand index, hoe stabieler de gevonden clusters zijn bij veranderingen in de dataset. Deze methode zal voor verschillende aantallen clusters bekeken worden om er zo achter te komen bij hoeveel clusters de allocatie van de dataset het meest stabiel is.

4. Simulatiestudie

Om alle bovenstaande methoden te vergelijken worden ze allemaal op een aantal gesimuleerde datasets toegepast en vergeleken aan de hand van de genoemde prestatiemaatstaven. In deze simulatie blijft het correcte aantal

clusters vastgesteld op vier en kijken we hoe de methoden presteren als ze verschillende aantallen clusters als input nemen. Voor elke simulatie worden vier variabelen gegenereerd met voor elk cluster een andere verdeling over de uitkomstenruimte. Er zijn per variabele vier categorien. Aangezien het juiste aantal clusters is vastgezet op vier worden er vier data-genererende processen gestart, voor elk cluster een. Binnen een data-genererend proces is de verdeling over de uitkomstenruimte hetzelfde voor alle observaties. Binnen een data-genererend proces is de verdeling ook hetzelfde voor alle variabelen. Dat wil zeggen de kans om bij de eerste variabele in de eerste categorie te vallen is evengroot als bij de tweede, derde of vierde variabele. Er worden drie mogelijke verdelingen bekeken met verschillende spreiding over de uitkomstenruimte. Deze verdelingen zijn als volgt:

1. 0.9, 0.05, 0.05, 0
2. 0.8, 0.1, 0.1, 0
3. 0.6, 0.25, 0.1, 0.05

In de resultaten zullen deze verdelingen worden aangeduid als respectievelijk 'prob1', 'prob2' en 'prob3'. Per situatie volgen alle clusters een verdeling met dezelfde kansen, alleen in een andere volgorde. Hieronder staat een voorbeeld hoe de clusters verdeeld zouden zijn als verdeling 1 van toepassing is. Dit voorbeeld laat de verdeling voor een variabele zien, maar zoals hierboven beschreven zijn de andere drie variabelen precies hetzelfde verdeeld voor alle clusters.

Table 1: Verdeling van een variabele voor vier clusters

	Categorie 1	catergorie 2	Categorie 3	Categorie 4
Cluster 1	0.9	0.05	0.05	0
Cluster 2	0	0.9	0.05	0.05
Cluster 3	0.05	0	0.9	0.05
Cluster 4	0.05	0.05	0	0.9

Naast de verschillende verdelingen worden de volgende factoren gevarieerd om hun invloed te kunnen bestuderen:

- formaat clusters: Er worden twee scenarios bekeken. In het eerste geval zijn alle clusters evengroot, dus beslaan ze een kwart van de observaties. In het tweede geval zijn de cluster van wisselende grootte, met verdeling 0.55, 0.25, 0.14, 0.06
- Aanwezigheid irrelevante variabelen: Er worden twee scenarios bekeken. In het eerste geval worden er alleen variabelen gebruikt die de cluster structuur verklaren, in het tweede geval worden er variabelen toegevoegd die niet bijdragen aan de cluster structuur. In dit geval is het aantal relevante en irrelevante variabelen gelijk.

Dit levert $2 \times 2 \times 3 = 12$ verschillende scenarios op met per scenario 1000 variabelen. Elk scenario wordt 20 keer gesimuleerd en bij elke simulatie

worden alle prestatiemaatstaven berekend behalve de methode van Dolnicar en Leisch. Voor deze methode wordt slechts een dataset gesimuleerd die vervolgens 50 keer gebootstrapt wordt. De gerapporteerde waarden zijn van alle prestatiemaatstaven het gemiddelde. Hoewel het correcte aantal clusters hetzelfde blijft worden alle methoden gebruikt met 2, 3, 4, 5, 6 en 7 clusters als input om te bekijken hoe de methoden dan presteren. Het aantal dimensies waartoe de data gereduceerd wordt is telkens voor de MCA k-means methode gelijk aan het aantal clusters waarnaar gezocht wordt en voor CCA 1 minder dan het aantal clusters waarnaar gezocht wordt. Alle simulaties zijn in Rstudio gedaan. Voor de CCA en MCA k-means methoden is gebruikgemaakt van de code uit Van de Velden et al. (2016). Voor de Silhouette score, Calinski-Harabasz index en Krzanowski-Lai index zijn respectievelijk de packages 'bios2mds', 'fpc' en 'clusterSim' gebruikt. Voor de bootstrap methode is de 'bootFlexClust' functie uit het package 'flexclust' gebruikt met een kleine aanpassing om te zorgen dat de gebootstrapte steekproeven met CCA of MCA k-means geclusterd worden in plaats van met k-means.

5. Resultaten

In de grafieken hieronder staan de resultaten van de simulatiestudie beschreven. Bij elke grafiek staat vermeld van welke methode en prestatiemaatstaf de resultaten zijn afgebeeld. De verschillend gekleurde lijnen geven de resultaten als naar verschillende aantallen clusters gezocht wordt (in de legenda aangegeven door 'k=' gevolgd door het aantal clusters). Onder de x-as van de grafiek staat voor welke situatie de prestatiemaatstaf de waarde bereikt die erboven in de grafiek staat afgebeeld. De woorden 'bal' en 'ong' staan voor 'gebalanceerd' en 'ongebalanceerd' en geven aan of alle clusters in de dataset evengroot zijn (gebalanceerd) of van verschillende grootte (ongebalanceerd). De woorden 'rel' en 'irr' staan voor 'relevant' en 'irrelevant' en geven aan of alleen de variabelen zijn meegenomen die de clusterstructuur bepalen (relevant) of dat er extra variabelen zijn toegevoegd die niet bijdragen aan de clusterstructuur (irrelevant). De woorden 'prob1', 'prob2' en 'prob3' verwijzen naar de in de vorige sectie beschreven verdelingen van de variabelen. Naast de grafieken zijn in de appendix tabellen te vinden waar alle waarden in te vinden zijn. In deze tabellen is per situatie en maatstaf de waarde dikgedrukt van het aantal clusters dat de maatstaf suggereert voor de dataset.

5.1. Silhouette score

Figure 1: Resultaten Silhouette score CCA

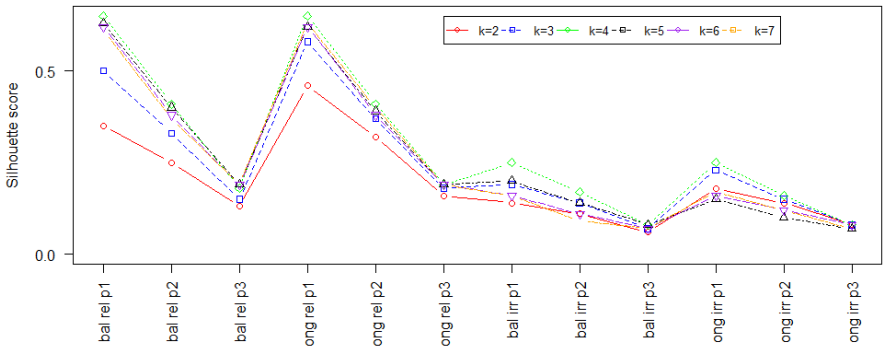
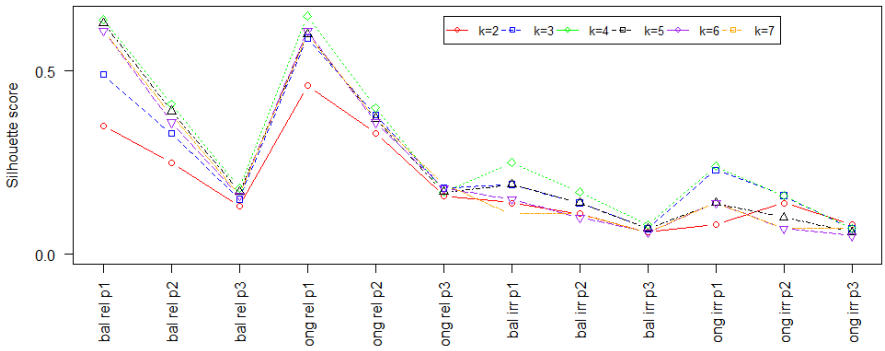


Figure 2: Resultaten Silhouette score MCA



De resultaten van de silhouettte score verschillen nauwelijks tussen de allocaties van de CCA en MCA k-means methode. Wat verder opvalt is de grote invloed van de verdeling van de variabelen op de silhouette score voor alle verschillen aantallen clusters waarnaar gezocht wordt. Ongeacht de grootte van het aantal clusters of de aanwezigheid van irrelevante variabelen neemt de silhouette score drastisch af naarmate de verdeling meer gespreid wordt over de dataset. Ook de aanwezigheid van irrelevante variabelen in de dataset blijkt van invloed op de uitkomst, maar de invloed van het aanpassen van de grootte van de clusters blijkt minder groot. Gevallen waarin alle eigenschappen hetzelfde zijn behalve de grootte van de clusters blijken vrijwel dezelfde resultaten op te leveren, terwijl dit niet het geval is voor gevallen waarbij de aanwezigheid

van irrelevante variabelen varieert. In vrijwel alle gevallen is de silhouette score het hoogst van de oplossing waarbij naar vier clusters gezocht wordt. Dit is het aantal clusters waaruit de dataset bij het simuleren is opgebouwd dus de silhouette score presteert over het algemeen vrij goed. Naarmate de verdeling meer gespreid wordt over de uitkomstenruimte komt de silhouette score van de verschillende oplossingen dichter bij elkaar en is minder duidelijk welk aantal clusters gesuggereerd wordt.

5.2. Calinski-Harabasz index

Figure 3: Resultaten Calinski-Harabasz index CCA

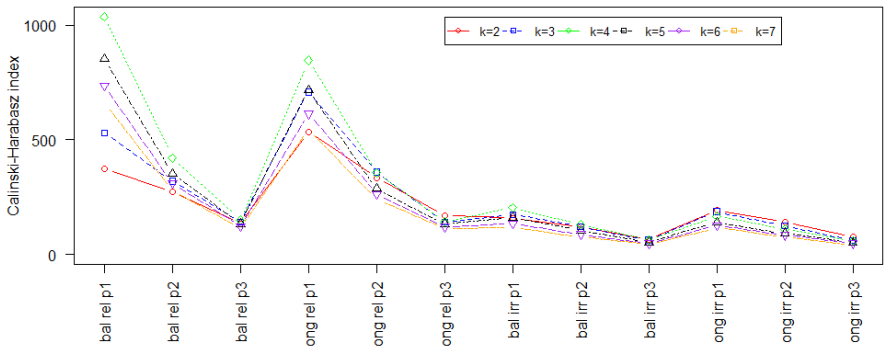
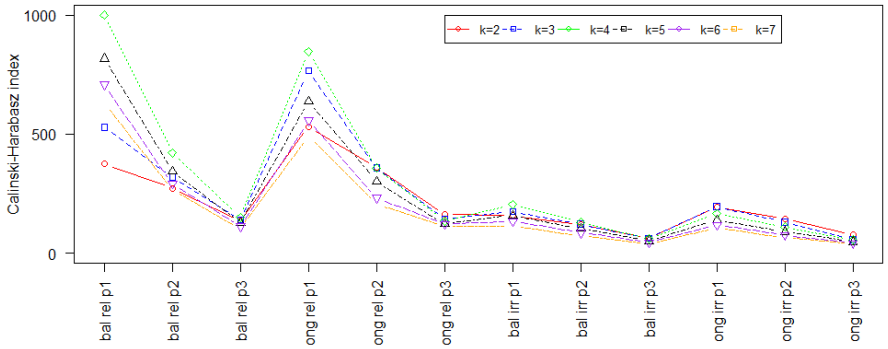


Figure 4: Resultaten Calinski-Harabasz index MCA



Ook als we naar de Calinski-Harabasz index kijken blijken de waardes weinig te verschillen tussen de oplossingen van de CCA en MCA k-means methoden. Ook blijkt hier net als bij de silhouette score dat het effect van de verdeling en

irrelevante variabelen goed te zien is, maar dat het effect van de grootte van de clusters minder evident is. Daarbij moet opgemerkt worden dat in vergelijking met de silhouette score de Calinski-Harabasz bij een meer gespreide verdeling en irrelevante variabelen aanzienlijk meer moeite heeft om het juiste aantal clusters te onderscheiden. In de relatief gemakkelijke gevallen met clusters van gelijke grootte en een niet al te gespreidde verdeling herkent de Calinski-Harabasz index duidelijk vier clusters in de dataset. Als dit niet het geval is liggen de waarden voor verschillende aantallen clusters echter erg dicht bij elkaar en wordt regelmatig een verkeerd aantal clusters gesuggereerd.

5.3. Krzanowski-Lai index

In de grafiek met de resultaten van de Krzanowski-Lai index zijn maar vier lijnen te zien, tegenover zes bij de andere prestatiemaatstaven. Dit komt omdat voor een bepaalde waarde van k de Krzanowski-Lai index de allocaties van de dataset in $k-1$, k en $k+1$ clusters vergelijkt. Aangezien alleen de oplossingen voor 2 tot en met 7 clusters zijn berekend is voor 2 en 7 clustrers geen Krzanowski-Lai index beschikbaar.

Figure 5: Resultaten Krzanowski-Lai index CCA

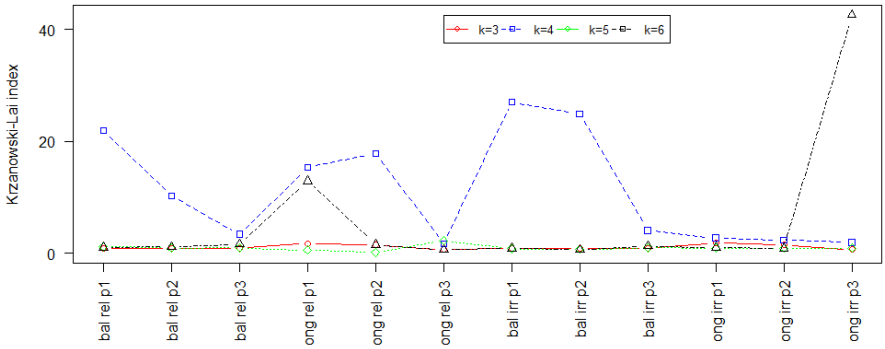
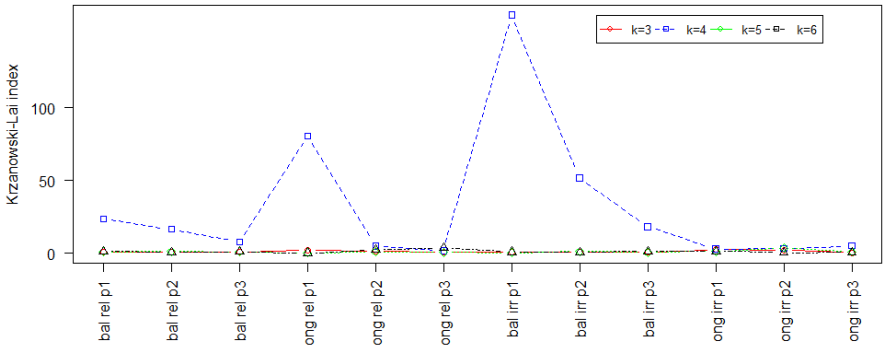


Figure 6: Resultaten Krzanowski-Lai index MCA



Het eerste dat opvalt is dat een paar uitkomsten (voornamelijk voor vier clusters) zoveel verschillen van de rest dat veel uitkomsten nauwelijks nog leesbaar zijn in de grafiek. Om de resultaten makkelijker leesbaar te maken zijn voor de grafieken hieronder van alle waarden de logaritmen genomen.

Figure 7: Resulaten logaritme Krzanowski-Lai index CCA

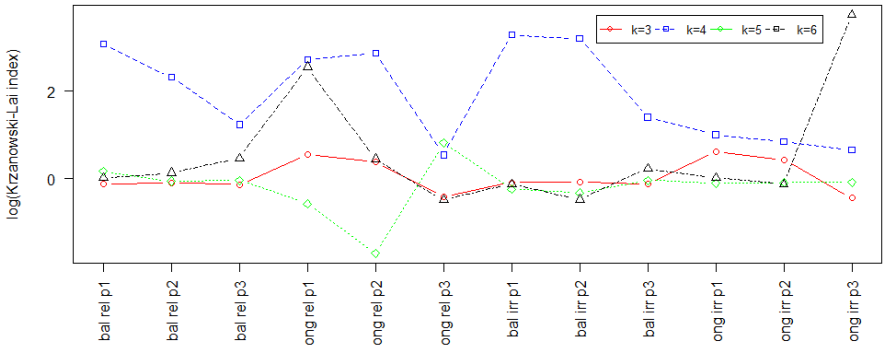
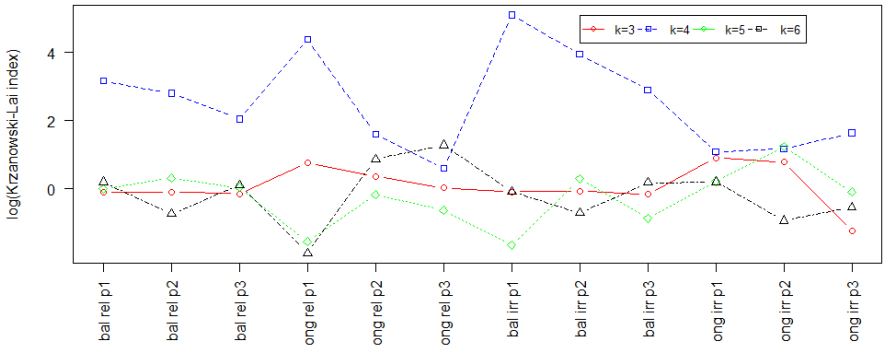


Figure 8: Resultaten logaritme Krzanowski-Lai index MCA



Kijkend naar de grafieken van de logaritmen valt op dat de uitkomsten van de CCA en MCA k-means meer verschillen dan het geval was bij de silhouette score en Calinski-Harabasz index. Het is verder lastig om de effecten van de grootte van de clusters of aanwezigheid van irrelevante variabelen te isoleren omdat bij elke combinatie van deze factoren de Krzanowski-Lai index duidelijk andere resultaten oplevert. Ook is het niet per se zo dat een meer gespreidde verdeling van de variabelen een lagere waarde van de Krzanowski-Lai index oplevert, wat bij de silhouette score en Calinski-Harabasz index vaak wel het geval was. Wel lijkt de Krzanowski-Lai index vrij consequent en ook vrij overtuigend het juiste aantal clusters te herkennen. Als dit niet het geval is hebben de variabelen vaak een meer gespreide verdeling. Bij het interpreteren van de uitkomsten is het belangrijk om in het achterhoofd te houden dat de Krzanowski-Lai index maar voor vier verschillende aantallen clusters is berekend en de andere prestatiemaatstaven voor zes. Dit maakt het voor de Krzanowski-Lai index waarschijnlijker dat het juiste aantal clusters wordt gesuggereerd. Gezien het grote verschil in veel situaties tussen de Krzanowski-Lai index voor vier clusters en alle andere aantallen is het moeilijk te zeggen hoeveel verschil het had gemaakt om voor meer waarden de Krzanowski-Lai index te berekenen.

Figure 9: Resultaten Dolnicar-Leisch bootstrap methode CCA

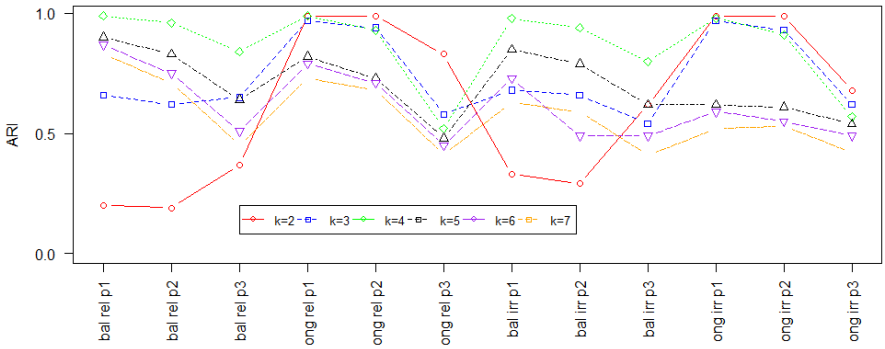
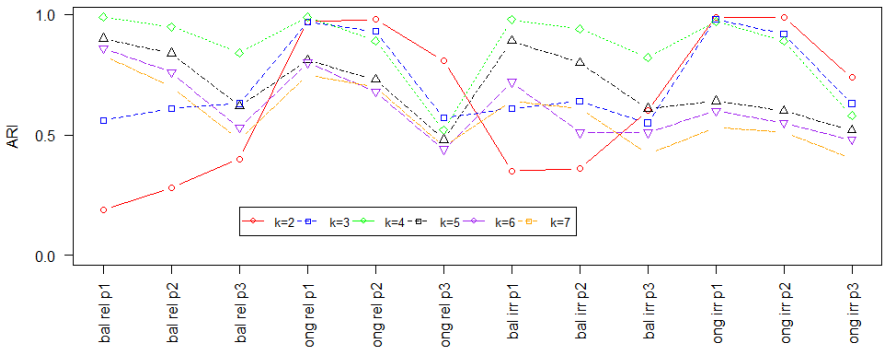


Figure 10: Resultaten Dolnicar-Leisch bootstrap methode MCA



De stabiliteit van de clusters blijkt tussen de CCA en MCA k-means methoden niet erg veel te verschillen. Wel lijkt in dit geval de invloed van de grootte van de clusters en de aanwezigheid van irrelevante variabelen anders dan eerder. Waar bij de silhouette score en de Calinski-Harabasz vooral de invloed van de irrelevante variabelen goed zichtbaar was en die van de grootte van de clusters minder is dat in dit geval eigenlijk andersom. In scenario's met clusters van gelijke groote gedraagt de uitkomst van de Dolnicar-Leisch bootstrap methode zich ongeveer hetzelfde ongeacht of er irrelevante variabelen in de dataset zitten, en hetzelfde geldt voor de scenario's waar de grootte van de clusters varieert. In vrijwel alle gevallen wordt de oplossing van de algoritmen minder stabiel als de verdeling meer gespreid wordt over de uitkomstenruimte.

Zolang de clusters evengroot zijn wordt voor elke verdeling van de clusters vrij overtuigend het juiste aantal clusters gesuggereerd, ongeacht of er irrelevante variabelen zijn toegevoegd. Als dit wel het geval is blijken de oplossingen met andere aantallen clusters soms stabieler, voornamelijk de oplossing met twee clusters.

6. Discussie

6.1. Conclusie

Het type dataset dat gesimuleerd wordt blijkt van grote invloed op de uitkomsten van de prestatiemaatstaven. In het simpelste geval waarin de clusters evengroot zijn en er geen irrelevante variabelen aan de dataset zijn toegevoegd is het vor alle prestatiemaatstaven vrij simpel om het juiste aantal clusters te onderscheiden. De vraag is dan ook gerechtvaardigd of dit wel een interessant geval was om te bekijken of dat deze data te simpel is. Het toevoegen van irrelevante variabelen maakt het voor de silhouette score en vooral de Calinski-Harabasz index moeilijker om het juiste aantal clusters te onderscheiden. Als de grootte van de clusters wordt gevarieerd suggereert de bootstrap methode van Dolnicar en Leisch vaker verkeerde aantallen clusters. Als de clusters in grootte varieren en er worden irrelevante variabelen toegevoegd is het voor alle prestatiemaatstaven moeilijk om het juiste aantal clusters te herkennen.

Al met al lijken de silhouette score en Krzanowski-Lai index het vaakst het juiste aantal clusters in de dataset te herkennen. Uiteraard moet hierbij in het achterhoofd gehouden worden dat de Krzanowski-Lai index minder waarden bekijkt en dus makkelijker het juiste aantal kan herkennen. Hierbij moet opgemerkt worden dat de prestaties van de Dolnicar-Leisch en Calinski-Harabasz index erg worden beinvloed door eigenschappen van de dataset. Als de Calinski-Harabasz een ander aantal clusters voorstelt dan de andere prestatiemaatstaven kan dit een indicatie zijn dat er irrelevante variabelen in de dataset zitten. Als de suggestie van de Dolnicar-Leisch index afwijkt kan dat een indicatie zijn dat de 'juiste' clusters in de dataset van verschillende grootte zijn.

6.2. Beperkingen

Er zijn een aantal zaken waar in dit paper niet of beperkt rekening mee zijn gehouden die van invloed kunnen zijn geweest op de resultaten. Bij vervolgstudies over dit onderwerp is het dan ook raadzaam om deze zaken specifieker te bekijken. Misschien wel de grootste beperking is dat het aantal dimensies waartoe de data wordt gereduceerd niet wordt bekeken. In de simulatiestudie in dit paper is ervan uitgegaan dat het aantal dimensies telkens gelijk was aan het aantal clusters voor MCA k-means en 1 minder dan het aantal clusters voor CCA. Hier was geen duidelijke motivatie voor en er is ook geen makkelijke vuistregel voorhanden die voorschrijft in welk aantal

dimensies de onderliggende clustering het best zichtbaar zal zijn. Het is voor vervolgonderzoek dus een interessant idee om per variatie ook het aantal dimensies te laten afwisselen en te kijken welk aantal de beste resultaten oplevert.

Een ander gebrek is dat het correcte aantal clusters in de onderliggende verdeling niet varieert, alleen het aantal clusters dat als input voor de CCA en MCAk-means methoden wordt gebruikt. Het is mogelijk dat het vormen van meer of minder clusters dan vier gevolgen heeft voor hoe goed de methoden ze terug kunnen vinden. Dit zou zeker het geval kunnen zijn als het aantal observaties hetzelfde blijft en het aantal observaties per cluster dus verandert. In dit geval zouden een aantal ongebruikelijke observaties in een cluster er sneller voor kunnen zorgen dat het cluster moeilijk te herkennen is in de data. Ook het aantal clusters dat als input voor de methoden wordt gebruikt beperkt natuurlijk enigszinds de resultaten, maar dit is waarschijnlijk minder significant. In de simulatiestudie zijn alleen twee tot en met zeven als input gebruikt. Een allocatie waar maar een enkel cluster gezocht wordt plaatst om triviale redenen alle observaties in hetzelfde cluster ongeacht de methode en is dus niet echt interessant om te analyseren. Voor acht of meer clusters wezen kleine experimenten uit dat er geen verbeteringen meer optreden in de prestatiemaatstaven die bekeken worden.

Een tekortkoming die eigenlijk niet te voorkomen is maar desondanks wel genoemd mag worden is dat uiteraard niet alle clusteringmethoden of prestatiemaatstaven zijn bekeken. In dit paper is speciaal gekeken naar methoden die zowel dimensie-reductie als clustering doen waarbij de meest triviale 'tandem' methode (correspondentie analyse gevolgd door k-means clustering) en i-FCB (D'enza et al., 2014) bijvoorbeeld niet zijn bekeken. Verder zijn er talloze prestatiemaatstaven, zowel intern als extern, die in dit paper niet behandeld zijn omdat het er simpelweg te veel zijn. Overzichten hiervan zijn te vinden in o.a. Liu et. al (2010), Milligan (1981) en Zhao en Franti (2014). Het kan als vervolg voor dit onderzoek interessant zijn om andere methoden te bekijken, zowel voor het clusteren van de data als voor het beoordelen van een gevonden allocatie.

7. Referenties

- Cordeiro de Amorim, R (2016). A survey on feature weighting based K-means algorithms. Journal of classification, 33, 210-242
- Dolnicar, S., Leisch, F.(2010). Evaluation of structure and reproducability of cluster solutions using bootstrap. Marketing letters, 21, 83-101.
- Iodice DEnza A., Van de Velden M., Palumbo F. (2014) On Joint Dimension Reduction and Clustering of Categorical Data. In: Vicari D., Okada A., Ragozini G., Weihs C. (eds) Analysis and Modeling of Complex Data in

Behavioral and Social Sciences. Studies in Classification, Data Analysis, and Knowledge Organization. Springer, Cham

- Efron, B., Tibshirani, R.J.(1993). An introduction to the bootstrap. New York: chapman and hall.
- Greenacre, M. J. (1984). Theory and applications of correspondence analysis. London: Academic Press.
- Hubert, L., Arabie, P.(1985). Comparing partitions. Journal of classification, 2, 193-218.
- Hwang, H., Dillon, W. R., & Takane, Y. (2006). An extension of multiple correspondence analysis for identifying heterogenous subgroups of respondents. Psychometrika, 71, 161171.
- Liu, Y., Li, Z., Xiong, H. Gao, X., Wu, J. (2010). Understanding of internal clustering validation measures. Bibliometrics, 911-916.
- Krzanowski, J., Lai, Y.T. (1988). A criterion for determining the number of groups in a dataset using sum-of-squares clustering. Biometrics, 44, 23-34.
- Milligan, G. W. (1981). A monte carlo study of thirty internatl criterion measures for clustering analysis. Psychometrika, 46, 187-199
- Mueen, A., Keogh, E. (2010). Online discovery and maintenance of time series motifs. Bibliometrics, 1089-1098
- Takane, Y., Young, F. & de Leeuw, J. (1976). Non-metric individual differences multidimensional scaling: an alternating least squares method with optimal scaling features. Psychometrika, 42, 767.
- van de Velden, M., D'enza, A. I. & Palumbo, F (2016). Cluster correspondence analysis. Psychometrika, 82, 158-185.
- Vichi, M., & Kiers, H. A. L. (2001). Factorial k-means analysis for two-way data. Computational Statistics and Data Analysis, 37, 49-64.
- Zhao, Q., Franti, P. (2014). WB-index: A sum-of-squares based index for cluster validity. Data & knowledge engineering, 92, 77-89

8. Appendix

Table 2: Resultaten Silhouette score CCA

	Prob1				Prob2				Prob3			
	Bal		Ong		Bal		Ong		Bal		Ong	
	Rel	Irr	Rel	Irr	Rel	Irr	Rel	Irr	Rel	Irr	Rel	Irr
2	0,35	0,14	0,46	0,18	0.25	0,11	0,32	0,14	0.13	0,06	0,16	0,08
3	0,5	0,19	0,58	0,23	0,33	0,14	0,37	0,15	0,15	0,07	0,18	0,08
4	**0,65**	**0,25**	**0,65**	**0,25**	**0,41**	**0,17**	**0,41**	**0,16**	0.18	**0,08**	**0,19**	**0,08**
5	0.63	0,20	0,62	0,15	0,40	0,14	0,39	0,10	0,19	0,08	0,19	0,07
6	0.62	0,16	0,62	0,16	0,38	0,11	0,38	0,12	0,19	0,07	0,19	0,08
7	0,61	0,16	0,63	0,17	0.37	0,09	0,40	0,12	**0,19**	0,07	0,19	0,07

Table 3: Resultaten Silhouette score MCA K-means

	Prob1				Prob2				Prob3			
	Bal		Ong		Bal		Ong		Bal		Ong	
	Rel	Irr	Rel	Irr	Rel	Irr	Rel	Irr	Rel	Irr	Rel	Irr
2	0,35	0,14	0,46	0,18	0.25	0,11	0,33	0,14	0.13	0,06	0,16	**0,08**
3	0,49	0,19	0,59	0,23	0,33	0,14	0,38	0,16	0,15	0,07	0,18	0,07
4	**0,64**	**0,25**	**0,65**	**0,24**	**0,41**	**0,17**	**0,40**	**0,16**	**0.18**	**0,08**	0,17	0,07
5	0,63	0,19	0,60	0,14	0,39	0,14	0,37	0,10	0,17	0,07	0,17	0,06
6	0,61	0,15	0,61	0,14	0,36	0,10	0,36	0,07	0,16	0,06	0,18	0,05
7	0,61	0,11	0,60	0,14	0,38	0,11	0,37	0,07	0,16	0,06	**0,19**	0,07

Table 4: Resultaten Calinski-Harabasz index CCA

	Prob1				Prob2				Prob3			
	Bal		Ong		Bal		Ong		Bal		Ong	
	Rel	Irr	Rel	Irr	Rel	Irr	Rel	Irr	Rel	Irr	Rel	Irr
2	373	158	534	**193**	274	121	335	**143**	135	**64**	**169**	**77**
3	530	174	708	187	320	122	**359**	125	141	63	142	61
4	**1036**	**204**	**846**	167	**421**	**130**	357	110	**151**	63	143	58
5	852	161	717	140	351	104	287	93	134	54	133	52
6	736	135	614	126	310	88	263	84	123	48	120	48
7	660	118	544	116	281	77	240	78	113	44	114	42

Table 5: Resultaten Calinski-Harabasz index MCA K-means

	Prob1				Prob2				Prob3			
	Bal		Ong		Bal		Ong		Bal		Ong	
	Rel	Irr	Rel	Irr	Rel	Irr	Rel	Irr	Rel	Irr	Rel	Irr
2	374	158	532	**194**	273	121	**360**	**145**	135	**64**	**162**	**77**
3	529	174	767	196	320	122	359	131	141	62	143	57
4	**1000**	**204**	**846**	167	**421**	**130**	357	110	**151**	62	135	56
5	817	159	637	138	343	103	300	90	128	52	122	48
6	708	133	557	119	293	86	232	75	113	45	123	42
7	631	115	492	105	265	74	207	66	102	40	112	40

Table 6: Resultaten Krzanowski-Lai index CCA

	Prob1				Prob2				Prob3			
	Bal		Ong		Bal		Ong		Bal		Ong	
	Rel	Irr	Rel	Irr	Rel	Irr	Rel	Irr	Rel	Irr	Rel	Irr
3	0,89	0,91	1,72	1,84	0,90	0,92	1,46	1,54	0,87	0,88	0,66	0,64
4	**21,88**	**27,06**	**15,36**	**2,73**	**10,28**	**24,86**	**17,78**	**2,33**	**3,44**	**4,10**	1,72	1,93
5	1,18	0,78	0,56	0,90	0,94	0,72	0,18	0,91	0,96	0,96	**2,26**	0,92
6	1,01	0,88	12,90	1,02	1,22	0,62	1,56	0,88	1,59	1,25	0,61	**42,57**

Table 7: Resultaten Krzanowski-Lai index MCA K-means

	Prob1				Prob2				Prob3			
	Bal		Ong		Bal		Ong		Bal		Ong	
	Rel	Irr	Rel	Irr	Rel	Irr	Rel	Irr	Rel	Irr	Rel	Irr
3	0,91	0,91	2,15	2,45	0,90	0,93	1,43	2,19	0,86	0,85	1,02	0,29
4	**23,72**	**163,64**	**80,83**	**2,95**	**16,45**	**51,79**	**4,96**	3,21	**7,69**	**18,16**	1,81	**5,13**
5	0,99	0,19	0,21	1,24	1,36	1,34	0,83	**3,45**	1,03	0,42	0,53	0,90
6	1,22	0,92	0,15	1,22	0,48	0,49	2,36	0,39	1,11	1,19	**3,59**	0,58

Table 8: Resultaten Dolnicar-Leisch bootstrap methoden CCA

	Prob1				Prob2				Prob3			
	Bal		Ong		Bal		Ong		Bal		Ong	
	Rel	Irr	Rel	Irr	Rel	Irr	Rel	Irr	Rel	Irr	Rel	Irr
2	0,20	0,33	0,99	**0,99**	0,19	0,29	**0,99**	**0,99**	0,37	0,62	**0,83**	**0,68**
3	0,66	0,68	0,97	0,97	0,62	0,66	0,94	0,93	0,65	0,54	0,58	0,62
4	**0,99**	**0,98**	**0,99**	0,98	**0,96**	**0,94**	0,93	0,91	**0,84**	**0,80**	0,52	0,57
5	0,90	0,85	0,82	0,62	0,83	0,79	0,73	0,61	0,64	0,62	0,48	0,54
6	0,87	0,73	0,79	0,59	0,75	0,49	0,71	0,55	0,51	0,49	0,45	0,49
7	0,83	0,63	0,73	0,52	0,71	0,59	0,68	0,53	0,45	0,41	0,41	0,42

Table 9: Resultaten Dolnicar-Leisch bootstrap methode MCA K-means

	Prob1				Prob2				Prob3			
	Bal		Ong		Bal		Ong		Bal		Ong	
	Rel	Irr	Rel	Irr	Rel	Irr	Rel	Irr	Rel	Irr	Rel	Irr
2	0,19	0,35	0,97	**0,99**	0,28	0,36	**0,98**	**0,99**	0,40	0,60	**0,81**	**0,74**
3	0,56	0,61	0,97	0,98	0,61	0,64	0,93	0,92	0,63	0,55	0,57	0,63
4	**0,99**	**0,98**	**0,99**	0,97	**0,95**	**0,94**	0,89	0,89	**0,84**	**0,82**	0,52	0,58
5	0,90	0,89	0,81	0,64	0,84	0,80	0,73	0,60	0,62	0,61	0,48	0,52
6	0,86	0,72	0,80	0,60	0,76	0,51	0,71	0,55	0,53	0,51	0,44	0,48
7	0,83	0,64	0,75	0,53	0,70	0,61	0,68	0,51	0,48	0,42	0,45	0,40