Faculteit Wetenschappen
Vakgroep Toegepaste Wiskunde en Informatica
Voorzitter: Prof. Dr. G. Vanden Berghe

Het Clusteren van Zoekresultaten met behulp van Vaagmieren

door

Steven Schockaert

Promotor: Prof. Dr. E..E. Kerre
Scriptiebegeleiders: Dr. C. Cornelis en Dr. M. De Cock

Scriptie ingediend tot het behalen van de academische graad van licentiaat in de informatica

Academiejaar 2003–2004

Voorwoord

Het werken aan deze scriptie gedurende het afgelopen academiejaar is voor mij een bijzonder boeiende en leerrijke ervaring geweest. Ik wens dan ook uitdrukkelijk mijn promotor prof. Kerre en mijn begeleiders Martine en Chris te bedanken voor de mogelijkheid die ze me gegeven hebben om rond een onderwerp te werken waarin ik me volledig heb kunnen vinden. Bovendien ben ik hen erg dankbaar voor de vrijheid waarmee ik heb kunnen werken, het vertrouwen dat ze in mij gesteld hebben, hun nooit aflatende enthousiasme en de vele uren die ze gespendeerd hebben aan het nauwgezet nalezen van deze scriptie. Ik hoop dan ook van harte op een verdere samenwerking de komende jaren. Voor de morele steun ben ik mijn ouders, goede vrienden en Tine in het bijzonder, erg dankbaar. I also would like to thank Dawid Weiss for his help with the Carrot^2 framework. The possibility of using Carrot^2 has really lifted a great weight off my shoulders.

Deze scriptie is verder als volgt ingedeeld. In hoofdstuk 1 worden enkele modellen voor het gedrag van echte mieren besproken vanuit een biologisch standpunt. Omwille van de uitgebreidheid van dit onderwerp, heb ik mij hierbij beperkt tot deze aspecten die reeds hun nut bij het ontwerpen van algoritmen bewezen hebben. Hoofdstuk 2 behandelt enkele bestaande clusteringsalgoritmen die op één of andere manier op dit gedrag gebaseerd zijn. De belangrijkste voordelen en beperkingen van deze algoritmen worden hierbij geïdentificeerd. In hoofdstuk 3 worden enkele wiskundige begrippen die nodig zijn voor de verdere hoofdstukken besproken. In het bijzonder worden vaagverzamelingen en -regels, ruwverzamelingen en formele conceptanalyse behandeld. Vaagregels worden in hoofdstuk 4 gebruikt bij de beschrijving van een nieuw miergebaseerd clusteringsalgoritme. Formele conceptanalyse, vaagverzamelingen en ruwverzamelingen worden gebruikt in hoofdstuk 5 waar nagegaan wordt hoe we zoekresultaten, beschouwd als documenten, kunnen vergelijken. Tot slot worden de bekomen resultaten in hoofdstuk 6 gecombineerd om te komen tot een algoritme voor het clusteren van zoekresultaten.

Steven Schockaert, mei 2004

Toelating tot bruikleen

"De auteur geeft de toelating deze scriptie voor consultatie beschikbaar te stellen en delen van de scriptie te kopiëren voor persoonlijk gebruik.
Elk ander gebruik valt onder de beperkingen van het auteursrecht, in het bijzonder met betrekking tot de verplichting de bron uitdrukkelijk te vermelden bij het aanhalen van resultaten uit deze scriptie."

Steven Schockaert, mei 2004

Het Clusteren van Zoekresultaten met behulp van Vaagmieren

door

Steven Schockaert

Scriptie ingediend tot het behalen van de academische graad van licentiaat in de informatica

Academiejaar 2003–2004

Promotor: Prof. Dr. E. E. Kerre
Scriptiebegeleiders: Dr. C. Cornelis en Dr. M. De Cock

Faculteit Wetenschappen
Universiteit Gent

Vakgroep Toegepaste Wiskunde en Informatica
Voorzitter: Prof. Dr. G. Vanden Berghe

Samenvatting

Algoritmen voor het clusteren van de resultaten van zoekmachines op het internet proberen de klassieke geordende lijst van resultaten om te vormen tot een, eventueel hiërarchische, structuur van clusters. Een dergelijke clusterstructuur kan een uitkomst bieden voor een gebruiker die geen concrete zoektermen kan bedenken voor de informatie die hij wenst; voor een gebruiker die op zoek is naar algemene informatie over een, typisch redelijk breed, onderwerp; voor zoekopdrachten waarbij de optredende zoektermen dubbelzinnig zijn, enz. We introduceren in deze scriptie een nieuw clusteringsalgoritme, geïnspireerd op het gedrag van mieren, voor het clusteren van dergelijke zoekresultaten. Dit vereist enerzijds een geschikt generiek clusteringsalgoritme en anderzijds de mogelijkheid om deze zoekresultaten op een gepaste wijze te vergelijken. Miergebaseerde clusteringsalgoritmen hebben het voordeel dat geen a priori informatie, zoals het aantal clusters of een initiële partitionering van de gegevensverzameling vereist is. Bovendien zijn deze algoritmen typisch efficiënt en robuust, en lenen ze zich gemakkelijk tot een parallelle implementatie.

Trefwoorden

Biologisch geïnspireerde algoritmen, clusteren, soft computing, internet

Inhoudsopgave

Hoofdstuk 1

Biologische modellen voor het gedrag van mieren

1.1 Inleiding

Doorheen de evolutie heeft de mens een zekere intelligentie ontwikkeld die hem in staat stelt om complexe problemen op te lossen. Ook mieren zijn erin geslaagd veel van de problemen waarmee ze geconfronteerd worden te overwinnen. In tegenstelling tot de mens, die zijn overleving dankt aan individuele capaciteiten, vertonen individuele mieren een heel eenvoudig gedrag. Door directe en indirecte interacties met elkaar en met de omgeving slagen zij er echter in collectief een complex gedrag te realiseren. Aan dit complexe collectieve gedrag, dat ontstaat als gevolg van eenvoudige interacties tussen eenvoudige individuen, werd de naam zwermintelligentie (swarm intelligence) gegeven, als tegenhanger van de individuele intelligentie die bijvoorbeeld bij mensen voorkomt.

De invloed van mieren op hun omgeving kan nauwelijks worden overschat. Zo is het totale gewicht van alle mieren van dezelfde grootte-orde als het gewicht van alle mensen [59] en het gewicht van de mieren in het Amazoneregenwoud bedraagt ongeveer een derde van de totale dierlijke biomassa ervan [41]. Naar schatting komen per hectare grond in het Amazoneregenwoud ongeveer 8 miljoen mieren voor; in de savanne van Ivoorkust bedraagt dit aantal zelfs 20 miljoen [41]. Een superkolonie bestaande uit 306 miljoen mieren met een nest dat zich uitstrekt over een oppervlakte van 2.7 km^2 werd geobserveerd [41]. Verschillende aspecten van deze insecten maken hun gedrag tot een populair onderzoeksdomein bij biologen:

- De kolonie kan beschouwd worden als een superorganisme waarbij de individuen overeenkomen met de cellen. De procedures die beschrijven hoe het gedrag van de individuen leidt tot de ontwikkeling van de kolonie (sociogenese) vertonen sterke gelijkenissen met de procedures die beschrijven hoe wijzigingen van individuele cellen de ontwikkeling van een organisme beïnvloeden (morfogenese). De organisatie bij mieren kan echter veel gemakkelijker bestudeerd worden dan de organisatie van een organisme [41].

- Mieren passen zich zonder veel moeilijkheden aan wanneer ze in een omgeving gebracht worden die verschillend is van hun natuurlijke omgeving. Dit laat in het bijzonder toe om het gedrag van mieren in een laboratorium te bestuderen [41].

- Er zijn ongeveer 9000 verschillende miersoorten beschreven. Naar schatting bestaan er zo'n 20000 met een soms sterk verschillend individueel of collectief gedrag [32]. Bo-

> vendien komen ze voor in zo goed als alle ecosystemen tussen de poolcirkels: de enige gebieden waar geen mieren voorkomen zijn Antarctica, IJsland, Groenland, Polynesië en enkele afgelegen eilanden in de Atlantische en Indische oceaan [41].

Theorieën i.v.m. zelforganisatie (self-organization), die ontstonden in de context van fysica en chemie om het ontstaan van macroscopische patronen uit processen en interacties op het microscopische niveau te verklaren [12], werden ook toegepast op sociale insecten zoals mieren om aan te tonen hoe hun complexe gedrag ontstaat uit directe of indirecte interacties tussen eenvoudige individuen [11, 75]. Hoewel reeds heel wat onderzoek verricht is naar zelforganiserende fenomenen in diverse wetenschappen zoals fysica, chemie, biologie, economie, sociologie, ecologie, ... beschikt men nog niet over een algemeen aanvaarde definitie van organisatie [33].

De ontstane inzichten hebben ook een invloed gehad op het ontwerp van intelligente systemen die gebruik maken van het gedecentraliseerde karakter van de zwermintelligentie, van de flexibiliteit (aanpassen aan veranderende omgevingen) en robuustheid (het falen van individuen resulteert niet in het falen van het systeem) en van de eenvoud van de agenten [12]. Om een dergelijk systeem te bouwen is kennis vereist over zowel het individuele gedrag van de agenten als over de noodzakelijke interacties om het gewenste globale gedrag te bekomen. Een mogelijke aanpak is het bestuderen van sociale insecten, vervolgens een model opstellen voor hun gedrag, de parameters van dit model aanpassen (waardoor het model eventueel biologisch niet langer relevant wordt) en niet-biologische componenten toevoegen [12]. We zullen met dit doel voor ogen in dit hoofdstuk enkele biologische modellen behandelen, waarvan we de toepassingen wat informatica betreft uitstellen tot het volgende hoofdstuk. Als gevolg van het gebrek aan een algemene theorie voor zelforganisatie, worden vele van de gehanteerde begrippen ontleend aan fysische theorieën zoals (nonequilibrium) thermodynamica en statistische mechanica [33]. Zo spreekt men over microscopische interacties en het resulterende macroscopische systeemgedrag, fase-overgangen (phase transitions), het gemiddelde vrije pad (mean free path),...

1.2 Stigmergie

De term stigmergie werd op het einde van de jaren '50 van de vorige eeuw ingevoerd door de Franse bioloog P.P. Grassé. Bij het bestuderen van de constructie van het nest van termieten merkte hij op dat dit proces niet gestuurd werd door gecentraliseerde controle, maar dat het de omgeving zelf was die de mieren aanzette tot werken. Hij formuleerde het als volgt [40]:

> La coordination des tâches, la régulation des constructions ne dépendent pas directement des ouvriers, mais des constructions elles-memes. L'ouvrier ne dirige pas son travail, il est guidé par lui. C'est à cette stimulation d'un type particulier que nous donnons le nom du stigmergie.

De term is afkomstig van het Griekse *stigma* (prikkel) en *ergon* (werk) en betekent letterlijk het aanzetten tot werken als gevolg van het resultaat van werk en komt in essentie neer op het ontstaan van een bepaald gedrag bij agenten als gevolg van wijzigingen in de lokale omgeving, ontstaan door vroeger gedrag. Het is tegenwoordig echter gebruikelijk om een ruimere definitie te hanteren [40]. In het bijzonder wordt met werk elke wijziging aan de omgeving, die veroorzaakt werd door het dier, bedoeld. Bovendien kan de stimulatie waarvan sprake beschouwd worden als [40]:

1. De keuze voor de actie die wordt uitgevoerd, wordt gewijzigd.

2. De parameters van de geselecteerde actie worden gewijzigd. Met parameters wordt hier bijvoorbeeld de duur, sterkte, frequentie , ... bedoeld.

3. Zowel de keuze van de actie als de gehanteerde parameters zijn dezelfde, maar het resultaat is gewijzigd als gevolg van een vorige actie.

De eerste twee interpretaties worden actieve stigmergie genoemd, de derde passieve stigmergie. Om stigmergie te ondersteunen moet de omgeving in staat zijn om lokaal gewijzigd te worden door de agenten. Bovendien moeten deze wijzigingen lang genoeg aanwezig kunnen blijven om het gedrag van de agenten te beïnvloeden. Synergie, van het Griekse synergos, wat letterlijk samen werken betekent, verwijst naar het samengestelde effect van twee of meer samenwerkende elementen [68]. Stigmergie kan dus beschouwd worden als een speciaal geval van synergie waarbij een van de samenwerkende elementen de omgeving is [68]. We geven tot slot nog enkele voorbeelden van stigmergie bij mieren en termieten [4]:

- Termieten bouwen hun nesten door kleine modderballetjes te maken en die op de grond en op elkaar te plaatsen. Deze modderballetjes worden bovendien voorzien van een hoeveelheid van een bepaald feromoon. In het begin worden de balletjes willekeurig op de grond geplaatst. Door de aanwezigheid van het feromoon zijn de termieten geneigd om nieuwe modderballetjes te leggen op plaatsen waar al dergelijke balletjes liggen. Op deze manier ontstaan stapels. Bovendien worden deze stapels, opnieuw door de aanwezigheid van het feromoon, naar elkaar toe gebouwd waardoor bogen ontstaan. Deze bogen vormen de basisstructuren waarmee het nest gebouwd wordt.

- Bij het zoeken naar voedselbronnen wordt door de mieren een hoeveelheid feromoon achtergelaten die afhankelijk is van o.a. de rijkdom van en de afstand tot de voedselbron. De kans dat een mier een bepaald spoor volgt is een (niet-lineaire) functie van de feromonenconcentratie. Aangezien er initieel geen feromonen aanwezig zijn, ontstaan willekeurige sporen waar (door de lage feromonenconcentraties) veel van wordt afgeweken. De goede sporen worden na verloop van tijd versterkt, de slechte verdwijnen (feromonen verdampen na een zekere tijd). Uiteindelijk wordt maar één spoor overgehouden, wat normaal gezien ook het beste spoor is.

- Sommige mieren hebben de neiging om de dode lichamen van nestgenoten bij elkaar te leggen op plaatsen die ver van het nest gelegen zijn. Oorspronkelijk worden dode lichamen op willekeurige plaatsen achtergelaten. Wanneer reeds dode lichamen aanwezig zijn, zullen de mieren met grote waarschijnlijkheid nieuwe dode lichamen bij de reeds aanwezige leggen. Uiteindelijk zijn alle lichamen bevat in 1 of 2 clusters.

We komen verder in dit hoofdstuk nog uitvoerig terug op de laatste twee voorbeelden. Verschillende eigenschappen van deze insecten zullen we echter onbesproken laten. We vermelden hier expliciet de constructie van het nest, het zoeken naar een geschikte plaats voor het nest, de invloed van de koningin en de verdediging van het nest.

1.3 Taakverdeling

Op elk moment moeten verschillende taken worden uitgevoerd door een mierenkolonie [59]:

- zoeken naar voedsel
- onderhoud en constructie van het nest
- verdediging van het nest
- verzorgen van de larven en het aanreiken van voedsel

Mechanismen om het werk te verdelen over de verschillende individuen van de kolonie zijn bijgevolg noodzakelijk. Flexibiliteit op kolonieniveau om zich aan te passen aan externe uitdagingen en interne perturbaties is een essentieel kenmerk van het verdelen van het werk bij sociale insecten [74]. Hoewel verschillende morfologische specialisaties aangetroffen worden, zullen verschillende groepen van individuen zich dynamisch specialiseren in een welbepaalde taak [59]. We veronderstellen dat er met iedere taak een bepaalde stimulus geassocieerd is, m.a.w. een kwantitatieve maat die de noodzaak van het uitvoeren van de taak weergeeft. Wanneer mieren op zoek gaan naar voedsel voor de larven zou het aantal hongerige larven bijvoorbeeld de stimulus kunnen zijn voor deze taak. Onderstellen we nu een kolonie van N mieren $m_1, m_2, \ldots, m_N$ en M taken $t_1, t_2, \ldots, t_M$, dan associëren we met elke mier m_i $(i = 1, \ldots, N)$ en elke taak t_j $(j = 1, \ldots, M)$ een zekere drempelwaarde θ_{ij}. Experimenteel onderzoek ondersteunt dit drempelwaarde-idee bij o.a. honingbijen en bepaalde miersoorten [12]. Noteren we met X_i de toestand van de mier m_i, waarbij $X_i = j$ wanneer de taak t_j wordt uitgevoerd en $X_i = 0$ wanneer geen enkele taak wordt uitgevoerd. We beschouwen voor de eenvoud enkel het geval waarbij slechts 1 taak in aanmerking komt. Onderstellen we de probabiliteit per tijdseenheid dat het niet-actieve individu m_i deze taak t_j begint uit te voeren, gelijk aan [12]

$$T_{\theta_{ij}}(s_j) = \frac{s_j^2}{s_j^2 + \theta_{ij}^2} \tag{1.1}$$

Hierbij is s_j de stimulus die geassocieerd is met de taak t_j. Een actief individu stopt met het uitvoeren van zijn taak met een vaste probabiliteit per tijdseenheid. Hoewel dit eenvoudige model heel wat gelijkenissen vertoont met experimenteel waargenomen gedrag, zijn er enkele ernstige tekortkomingen die alle te wijten zijn aan het feit dat de drempelwaarden constant verondersteld worden over het bestudeerde tijdsinterval [74]: het model verklaart niet hoe de drempelwaarden tot stand komen en laat geen specialisatie toe op een robuuste manier; het model is wegens de constante drempelwaarden slechts geldig over een voldoende kort tijdsinterval en het model is niet consistent met meer recente experimenten bij honingbijen. Er wordt verondersteld dat zowel ouderdom als ervaring een invloed hebben op de specialisatie van individuen binnen een kolonie. Zo gaan de meeste mieren naarmate ze ouder worden over van het verzorgen van de larven naar het zoeken van voedsel [41]. Een dergelijke vorm van specialisatie wordt temporeel polyethisme genoemd. Experimenteel werd echter vastgesteld dat het vooral de relatieve ouderdom ten opzichte van de rest van de kolonie is die een rol speelt [74]. Dit alles suggereert dat een model met variabele drempelwaarden beter geschikt is voor het verklaren van de taakverdeling binnen een kolonie.

We beschouwen nu een uitbreiding van dit model waarbij de drempelwaarden aangepast worden. Wanneer een individu een taak uitvoert, wordt de corresponderende drempelwaarde verlaagd zodat dit individu in de toekomst sneller deze taak zou uitvoeren. Zij nu ξ en ϕ constanten die respectievelijk het leren en het vergeten beschrijven. Wanneer individu m_i de taak t_j heeft uitgevoerd tijdens het interval Δt wordt de drempelwaarde aangepast als [74]:

$$\theta_{ij} \to \theta_{ij} - \xi \Delta t$$

Wanneer de taak j niet werd uitgevoerd tijdens Δt wordt de drempelwaarde aangepast als [74]

$$\theta_{ij} \to \theta_{ij} + \phi\Delta t$$

We kunnen deze laatste twee uitdrukkingen nu combineren tot één uitdrukking als volgt. Noemen we x_{ij} de fractie van het interval Δt dat individu i gespendeerd heeft aan taak j. Dan krijgen we [74]

$$\theta_{ij} \to \theta_{ij} - x_{ij}\xi\Delta t + (1 - x_{ij})\phi\Delta t \tag{1.2}$$

Hierbij wordt θ_{ij} beperkt tot het interval $[\theta_{min}, \theta_{max}]$. In continue tijd kunnen we dit uitdrukken als

$$\frac{d\theta_{ij}}{dt} = ((1 - x_{ij})\phi - x_{ij}\xi)\ \Theta(\theta_{ij} - \theta_{min})\,\Theta(\theta_{max} - \theta_{ij})$$

met $\Theta(y) = 0$ als $y \leq 0$ en $\Theta(y) = 1$ voor $y > 0$. Het gedrag van x_{ij} in de tijd wordt beschreven m.b.v. [74]

$$\frac{dx_{ij}}{dt} = T_{\theta_{ij}}(s_j)\left(1 - \sum_{k=1}^{M} x_{ik}\right) - px_{ij} + \Psi(i, j, t)$$

met $\Psi(i, j, t)$ een normaal verdeeld stochastisch proces dat uitdrukt dat verschillende individuen de lokale toestand op een licht verschillende manier ervaren en p de vaste probabiliteit waarmee actieve individuen hun taak stopzetten. Voor de eenvoud werd p hier constant en identiek voor alle taken en individuen verondersteld. De evolutie van de stimulus in de tijd kunnen we beschrijven a.d.h.v. [74]

$$\frac{ds_j}{dt} = \delta - \frac{\alpha}{N}\left(\sum_{i=1}^{N} x_{ij}\right)$$

We veronderstellen hierbij dat de stimulus stijgt met een constante δ per tijdseenheid. α is een schaalfactor die de efficiëntie van de uitvoering van de taak voorstelt die we ook constant en identiek voor alle individuen en taken veronderstellen. De factor $1/N$ drukt uit dat de vraag (bijvoorbeeld naar voedsel) een stijgende functie is van de grootte van de kolonie die hier lineair verondersteld werd. Wanneer de grootte van de kolonie verdubbelt, zal de vraag naar voedsel inderdaad ook verdubbelen.

Tot slot merken we nog op dat om tot een betrouwbaar model te komen, de waarde voor alle optredende parameters experimenteel zal moeten worden vastgesteld. Deze parameterwaarden kunnen bovendien verschillen van soort tot soort.

1.4 Feromonensporen

Veruit de meest gebruikte vorm van communicatie bij mieren is het afscheiden van chemische stoffen die feromonen genoemd worden. Deze feromonen zijn vluchtig en de detectie ervan gebeurt in de lucht met behulp van twee antennes. De mier zal het spoor op een zigzaggende manier volgen zodat het feromoon afwisselend met de linker- en rechterantenne wordt waargenomen [32]. Een eerste klasse van feromonen worden primer-feromonen genoemd. Deze feromonen wijzigen het endocrinaal stelsel en het voortplantingsstelsel fysiologisch. Het lichaam van de mier wordt op deze manier gewijzigd voor een nieuwe biologische activiteit [41]. Een tweede klasse van feromonen worden releaser-feromonen genoemd. Hun werking

komt overeen met een stimulus-respons mechanisme dat volledig wordt afgehandeld door het zenuwstelsel [41]. Deze releaser-feromonen worden voor verschillende doeleinden gebruikt [32]:

- Alarmferomonen dienen voor het mobiliseren van soortgenoten. Wanneer mieren een ander nest overvallen, hebben deze feromonen bovendien nog een tweede doel. Ze worden gebruikt als een chemisch wapen om hun tegenstanders mee te bespuiten, met chaos en verwarring bij de tegenstanders tot gevolg. Ten minste één miersoort slaagt er op deze wijze in de tegenstanders elkaar te laten aanvallen.

- Communicatie over de toestand van het broed.

- Wanneer een mier sterft wordt een chemische stof, die oliezuur genoemd wordt, afgescheiden. Dit oliezuur doet dienst als feromoon en is voor de andere mieren het teken dat het kadaver uit het nest moet verwijderd worden. Uit experimenten met bijen waarbij o.a. een levende koningin met oliezuur werd overgoten, bleek dat zelfs tegenspartelende bijen uit het nest werden verwijderd.

- Veruit het belangrijkste gebruik van feromonen betreft spoorferomonen. Deze werden door Wilson [41] ontdekt rond 1950 en zorgen ervoor dat de mieren de weg naar een voedselbron terugvinden. Verkenners die een voedselbron gevonden hebben, laten bij het terugkeren naar het nest een feromonenspoor achter. In tegenstelling tot andere feromonen, moeten deze spoorferomonen geheim gehouden worden voor andere miersoorten. De chemische structuur ervan is bijgevolg sterk verschillend bij verschillende miersoorten. Mieren zijn heel gevoelig voor dit spoorferomoon, zo werd berekend dat één milligram ervan volstaat om een mierenkolonne drie maal rond de aarde te leiden.

Deze vorm van communicatie wordt massacommunicatie genoemd en laat toe complexe vormen van informatie over te brengen die geen enkel individu kent of kan overdragen [14]. Het construeren van feromonensporen is in zekere zin analoog als wat bij de werking van de hersenen cognitieve schema's (cognitive maps) genoemd worden [14]. Neurotransmitters komen dan overeen met feromonen en neuronen met mieren. Deze analogie is echter veel meer dan een leuke metafoor. Zo blijken neurotransmitters en feromonen ook sterke fysiologische gelijkenissen te vertonen. In de volgende sectie zullen we beschrijven hoe mieren aan de hand van deze spoorferomonen erin slagen om op een efficiënte manier voedselbronnen te vinden en de weg ernaar te onthouden.

We gaan nu eerst dieper in op een eenvoudig model van Chialvo en Millonas [14, 69]. Er wordt gezocht naar een zo eenvoudig mogelijk model dat ons in staat stelt de constructie van feromonensporen en het geassocieerde georganiseerde gedrag beter te begrijpen. Dit is essentieel gezien de grote praktische problemen die optreden bij het observeren van feromonen. Aangezien op fenomenologisch vlak heel wat ruis optreedt bij het volgen van feromonen, zal een beschrijving in ieder geval stochastisch van aard moeten zijn. De toestand van elke mier bestaat uit de positie r en de oriëntatie θ. De kans dat een mier een bepaald spoor volgt, is afhankelijk van de feromonendichtheid σ op dat spoor (in de onmiddellijke omgeving van de mier) en wordt evenredig ondersteld met [14]

$$W(\sigma) = \left(1 + \frac{\sigma}{1 + \delta\sigma}\right)^{\beta} \tag{1.3}$$

Hierbij is β een parameter die geassocieerd is met de gevoeligheid van de mier voor de feromonen en kan gezien worden als een inverse ruisparameter. Deze parameter wordt ook nog de

osmotropotaxische sensitiviteit genoemd (osmotropotaxis is het volgen van feromonensporen [69]). Hoe lager de waarde van β, hoe meer willekeur er in het gedrag van de mier zal aanwezig zijn. Verder wordt $1/\delta$ de sensorcapaciteit genoemd, deze beschrijft het feit dat het vermogen van de mier om feromonen waar te nemen ietwat afneemt voor hoge concentraties. Bovendien moet rekening gehouden worden met een gewichtsfactor $w(\Delta\theta)$ waarbij $\Delta\theta$ het verschil tussen de huidige en de nieuwe oriëntatie voorstelt. De precieze vorm van deze gewichtsfactor is niet of onvoldoende gekend. Noem nu η de hoeveelheid feromonen die de mieren achterlaten per tijdseenheid en κ de hoeveelheid feromonen die verdampt per tijdseenheid. De feromonen geven dus informatie over voorbije bewegingen, maar niet over bewegingen die willekeurig lang geleden zijn. Zij ρ_0 de gemiddelde dichtheid van de mieren, we kunnen dan de parameters ρ_0, η en κ combineren tot 1 parameter $\sigma_0 = \rho_0\eta/\kappa$ die we het gemiddelde feromonenveld noemen. Door de beweging van de mieren in het feromonenveld te beschrijven als een stochastische bewegingsvergelijking van het Langevin type en hierop mean-field stabiliteitsanalyse toe te passen kan worden aangetoond [69] dat geordend gedrag optreedt wanneer voldaan is aan

$$\frac{\sigma_0 f'(\sigma_0)}{f(\sigma_0)} - \frac{1}{\beta} > 0 \tag{1.4}$$

Hierbij is $f(\sigma) = 1 + \frac{\sigma}{1+\delta\sigma}$ en dus $f'(\sigma) = \frac{1}{(1+\delta\sigma)^2}$. We krijgen dus achtereenvolgens

$$\begin{aligned}
& \frac{\sigma_0 f'(\sigma_0)}{f(\sigma_0)} - \frac{1}{\beta} > 0 \\
\Leftrightarrow \quad & \frac{\sigma_0}{(1+\delta\sigma_0)^2(1+\frac{\sigma_0}{1+\delta\sigma_0})} - \frac{1}{\beta} > 0 \\
\Leftrightarrow \quad & \frac{\sigma_0}{(1+\delta\sigma_0)(1+\delta\sigma_0+\sigma_0)} - \frac{1}{\beta} > 0 \\
\Leftrightarrow \quad & \beta > \frac{1}{\sigma_0}(1+\delta\sigma_0)(1+\delta\sigma_0+\sigma_0) \\
\Leftrightarrow \quad & \beta > 1 + \frac{1}{\sigma_0} + 2\delta + \delta\sigma_0 + \delta^2\sigma_0
\end{aligned}$$

Deze transitie kan dus theoretisch bepaald worden. De resulterende patronen kunnen echter enkel d.m.v. simulatie bepaald worden. Om dit gedrag te simuleren, worden zowel tijd als plaats gediscretiseerd. We beschouwen m.a.w. mieren die zich op een vakje van een 2-dimensionaal rooster bevinden en zich elke (discrete) tijdstap naar 1 van de naburige vakjes begeven. Op elk vakje kan een hoeveelheid feromonen aanwezig zijn. Noteren we met σ_i de hoeveelheid feromonen op het vakje i. Om effecten die veroorzaakt worden door de randen van het rooster te vermijden, zullen we gebruik maken van een toroïdaal rooster (de linker- en rechterrand en de onder- en bovenrand van het rooster zijn met elkaar verbonden). De hoeveelheid feromonen die aanwezig is op het eigen vakje en op de naburige vakjes kan worden waargenomen. De probabiliteit dat van vakje k naar het naburige vakje i gegaan wordt, wordt gegeven door [14, 69]

$$P_{ik} = \frac{W(\sigma_i)w(\Delta_i)}{\sum_{j\in N(k)} W(\sigma_j)w(\Delta_j)}$$

waarbij $N(k)$ gedefinieerd wordt als de verzameling bestaande uit de acht naburige vakjes op het rooster en $W(\sigma_i)$ nog steeds gegeven wordt door (1.3). Gezien de roostervoorstelling, kan

de oriëntatie van een mier slechts 8 waarden aannemen, die we zouden kunnen aanduiden als noord, noord-oost, oost, ... Bijgevolg kan het verschil in oriëntatie Δ_i enkel de waarden $0, 1, 2, 3$ en 4 kan aannemen. De gewichtsfunctie w moet dus enkel voor deze waarden gespecificeerd worden.

De voorwaarde (1.4) verdeelt de $(\sigma_0, \beta, \delta)$-parameterruimte in twee gebieden. Het gebied waar de punten aan de voorwaarde voldoen, zullen we het georganiseerd gebied noemen. Uit simulaties [14] blijkt dat voor waarden van de parameters die in het georganiseerd gebied gelegen zijn, de mieren na een klein (bijvoorbeeld 100) aantal stappen effectief georganiseerd gedrag beginnen te vertonen. Er onstaat een netwerk van sporen waarover de mieren bewegen. Wanneer de parameters zich echter te ver van de overgangswaarden bevinden (maar nog steeds in het georganiseerd gebied), ontstaan geen lijnen meer, maar enkel lokale lussen waarbij de mieren steeds hun eigen spoor blijven volgen m.b.v. U-bochten. Voor parameterwaarden buiten het georganiseerd gebied bewegen de mieren willekeurig zonder sporen te vormen.

De flexibiliteit die ontstaat bij dit model moge blijken uit de volgende experimenten. In een eerste experiment [14] werd gestart met parameters in het georganiseerd gebied zodat de mieren sporen vormden. Vervolgens werd de parameter β verminderd zodat willekeurig gedrag ontstond en grote fluctuaties optraden in de feromonendistributie. Wanneer vervolgens de parameter β in zijn oorspronkelijke toestand hersteld werd, werden zo goed als exact dezelfde sporen bekomen als voor het wijzigen van β. In een tweede experiment [69] werden parameterwaarden buiten, maar niet te ver van, het georganiseerd gebied genomen. Bovendien werd het rooster geïnitialiseerd met een spoor dat bestond uit een kleine hoeveelheid feromonen. Hoewel de parameterwaarden buiten het georganiseerd gebied gelegen waren, ontstond toch georganiseerd gedrag (de mieren volgden de sporen waarmee het gebied geïnitialiseerd werd). Dit gedrag zou echter niet spontaan ontstaan zijn.

We kunnen hieruit het volgende concluderen. Aangezien alle patronen die ontstaan initieel gebaseerd zijn op een zwakke structuur kan de groep zich gedragen als een informatieversterker waarbij zelfs zwakke externe verstoringen (zoals bijvoorbeeld de aanwezigheid van een voedselbron) tot een significante reactie kunnen leiden [14]. Parameterwaarden die tot een flexibel en adaptief georganiseerd gedrag leiden zullen dus in het georganiseerd gebied gelegen zijn, maar niet te ver van de overgang. We merken tot slot nog de gelijkenis op met de "rand van de chaos" hypothese. Deze hypothese stelt dat er in de parameterruimte van een dynamisch systeem gebieden zullen bestaan waar eenvoudig gedrag wordt aangetroffen en andere gebieden waar chaotisch gedrag wordt aangetroffen. Bij de overgangen tussen deze gebieden zal interessanter gedrag worden aangetroffen dat noch eenvoudig, noch chaotisch van aard is [35].

1.5 Zoeken naar voedsel

Het zoeken naar en ophalen van voedsel wordt meestal foerageren (foraging) genoemd [32], een taak die vaak gepaard gaat met grote verliezen voor de kolonie [41]. Bij sommige soorten gaan individuele mieren op jacht. Andere soorten, de zogenaamde legermieren (army ants) jagen in groep op grotere prooien die ze vervolgens gezamenlijk naar het nest dragen [41]. We bespreken hier een andere situatie waarbij verkenners op zoek gaan naar voedselbronnen. Wanneer ze deze ontdekken, delen ze de locatie ervan mee aan de soortgenoten door gebruik te maken van spoorferomonen. In het model van Chialvo en Millonas in de vorige sectie hebben we verondersteld dat de mieren steeds een constante hoeveelheid feromonen achterlaten. In

de natuur zal dit echter niet het geval zijn. Verkenners die op zoek gaan naar voedselbronnen zullen normaal gezien geen feromonen achterlaten tot ze een voedselbron gevonden hebben. Wanneer ze een voedselbron ontdekt hebben, zullen ze enkel bij het terugkeren naar het nest een spoor achterlaten. Mieren die beslissen het spoor te volgen zullen dit spoor versterken door opnieuw feromonen achter te laten. Merk op dat dit wegens het vluchtig zijn van de feromonen noodzakelijk is om het pad naar de voedselbron niet te verliezen. Door het vluchtige karakter van de feromonen zullen voedselbronnen die dichtbij gelegen zijn sneller een sterk spoor kunnen ontwikkelen dan veraf gelegen voedselbronnen.

1.5.1 Invloed van de koloniegrootte

Veronderstellen we nu eerst dat er slechts één voedselbron aanwezig is. Noem N het totale aantal mieren in de kolonie en x het aantal mieren dat de voedselbron bezoekt en dus het spoor volgt. De kans dat een willekeurige mier de voedselbron begint te bezoeken is afhankelijk van de kans dat deze mier de voedselbron vindt door zelfstandig te zoeken en van de kans dat ze naar de voedselbron geleid wordt door het aanwezige feromonenspoor. De probabiliteit per tijdseenheid dat een mier de voedselbron zelfstandig vindt, kunnen we constant veronderstellen en noemen we α [5]. De probabiliteit dat een mier naar de voedselbron geleid wordt door het aanwezige feromonenspoor is afhankelijk van de sterkte van dit spoor of equivalent hiermee, van het aantal mieren dat reeds dit spoor volgt. We kunnen deze laatste probabiliteit gelijk stellen aan βx, waarbij β een constante is [5]. Mieren die een spoor volgen zullen dit niet gegarandeerd blijven doen. De probabiliteit dat een mier het spoor verliest, stellen we gelijk aan $sx/(s+x)$ met s een constante [5]. De constanten α, β en s hangen af van de topologie van de zoekruimte en de eigenschappen van de beschouwde miersoort en moeten experimenteel worden vastgesteld. De snelheid waarmee x gewijzigd wordt, wordt dan gegeven door

$$\frac{dx}{dt} = (\alpha + \beta x)(N - x) - \frac{sx}{s + x} \tag{1.5}$$

Merk op dat we hierbij verondersteld hebben dat alle mieren die het spoor niet volgen, op zoek zijn naar voedselbronnen. Als alternatief voor de laatste term in het rechterlid wordt soms ook $sx/(x+c)$ gebruikt, met c een nieuwe constante [72, 73]. Door de nulpunten van het rechterlid in (1.5) te bepalen, bekomen we de evenwichtstoestanden (steady states). Een evenwichtstoestand wordt stabiel (resp. onstabiel) genoemd als een kleine verstoring in de evenwichtstoestand zal uitdoven (resp. versterkt worden) [73]. Noem $f = \frac{dx}{dt}$, m.a.w. zij $f(x)$ gedefinieerd als het rechterlid van (1.5). Een evenwichtstoestand x_* zal dan stabiel zijn als $\frac{df}{dx}(x_*) < 0$. We kunnen dit gemakkelijk inzien door de eerste 2 termen van de Taylorreeks van f rond x_* te beschouwen:

$$f(x_* + y) = f(x_*) + yf'(x_*) = y\frac{df}{dx}(x_*)$$

Afhankelijk van de parameterwaarden zullen voor differentiaalvergelijking (1.5) 1 of 3 evenwichtstoestanden bestaan [5]. In het geval er slechts 1 evenwichtstoestand is, zal deze stabiel zijn [5]. Het totaal aantal mieren dat het spoor uiteindelijk zal volgen is uniek bepaald. Het verband tussen de koloniegrootte en het uiteindelijke aantal mieren op het spoor is continu, in die zin dat een kleine wijziging in de koloniegrootte een kleine wijziging op het aantal mieren op het spoor tot gevolg heeft. Men spreekt in dit geval van een tweede-orde faseovergang. Wanneer er echter 3 evenwichtstoestanden zijn, zullen 2 ervan stabiel zijn en 1 onstabiel [5].

Het optreden van een onstabiele evenwichtstoestand impliceert dat het uiteindelijke aantal mieren dat het spoor zal volgen afhankelijk is van het initiële aantal mieren op het spoor. Men spreekt in dit geval van hysteresis. Als initieel minder (resp. meer) mieren dan het aantal dat correspondeert met de onstabiele toestand het spoor volgen, zal geëvolueerd worden naar een stabiele toestand die correspondeert met een kleiner (resp. groter) aantal dan de onstabiele toestand. Aangezien dit initiële aantal ook afhangt van de grootte van de kolonie, is het verband tussen de koloniegrootte en het uiteindelijke aantal mieren op het spoor hier discontinu. Men spreekt dan over een eerste-orde faseovergang.

Experimentele resultaten in [5] m.b.v. faraomieren (u wellicht bekend uit de keuken) tonen aan dat zelfs een kolonie van 600 mieren niet volstaat om georganiseerd gedrag te vertonen m.b.t. het ophalen van voedsel dat zich 50 cm van het nest bevindt. Dit wordt hoofdzakelijk toegeschreven aan het vluchtige karakter van de feromonen. Hoewel sommige miersoorten gebruik maken van feromonen die minder vluchtig zijn om bijvoorbeeld belangrijke voedselbronnen aan te geven, leidt dit tot een beperking van het adaptief gedrag. Bijgevolg zijn dergelijke feromonen niet geschikt voor tijdelijke voedselbronnen. Mieren hebben verschillende mechanismen ontwikkeld om tegemoet te komen aan deze noodzaak voor een grote kolonie. Zo laten sommige miersoorten ook een feromonenspoor achter wanneer ze nieuwe gebieden verkennen, zelfs wanneer ze (nog) geen voedselbron gevonden hebben. Dit heeft een stijging van de lokale densiteit van de mieren tot gevolg, zodat wanneer een voedselbron gevonden wordt, de probabiliteit dat een spoor gevormd wordt vergroot. Andere miersoorten maken gebruik van groeprecrutering. Wanneer een mier in dit geval een voedselbron gevonden heeft, leidt hij rechtstreeks een aantal andere mieren tot aan de voedselbron (wanneer dit aantal beperkt blijft tot 1 mier, spreekt men van tandemrecrutering). Aangezien deze groeprecrutering niet beperkt is door de koloniegrootte, vinden we deze techniek vooral terug bij soorten met een kleine kolonie. Sommige soorten maken gebruik van zowel groeprecrutering als feromonensporen [5].

1.5.2 Zoeken naar de optimale voedselbron

Hoewel het duidelijk is dat de mieren aan de hand van spoorferomonen hun ervaringen kunnen delen en bijgevolg door samenwerking sneller een voedselbron kunnen vinden, is het heel wat minder duidelijk dat mieren op deze manier de beste voedselbron kiezen [72]. We beschouwen nu het geval waarbij er 2 voedselbronnen aanwezig zijn. Veralgemening naar meer voedselbronnen ligt dan voor de hand. We beperken ons bovendien tot het geval waarbij enkel feromonensporen gebruikt worden. Hoewel sommige mieren beide bronnen zullen bezocht hebben, heeft het merendeel van de mieren niet de kans gehad de bronnen te vergelijken [72]. De verdeling van de mieren over de verschillende bronnen ontstaat door communicatie via de feromonensporen. Noem x_a het aantal mieren dat de eerste voedselbron bezoekt en x_b het aantal dat de tweede bron bezoekt. De koloniegrootte stellen we nog steeds voor door N en we veronderstellen dat de $N - x_a - x_b$ overblijvende mieren alle op zoek zijn naar voedselbronnen. Voor de eenvoud veronderstellen we dat de 2 bronnen even ver van het nest gelegen zijn. We kunnen het alternatief voor differentiaalvergelijking (1.5) dan eenvoudig uitbreiden naar het geval van 2 bronnen als [72]

$$\begin{aligned} \frac{dx_a}{dt} &= (\alpha + \beta_a x_a)(N - x_a - x_b) - \frac{s x_a}{c + x_a} \\ \frac{dx_b}{dt} &= (\alpha + \beta_b x_b)(N - x_a - x_b) - \frac{s x_b}{c + x_b} \end{aligned}$$

β_a en β_b bepalen de sterkte van de recrutering. Deze sterkte is afhankelijk van de hoeveelheid feromonen op het spoor. Die hoeveelheid feromonen op het spoor is op zijn beurt afhankelijk van de sterkte van de voedselbron, als we veronderstellen dat op het spoor naar een zwakke voedselbron, de individuele mieren een kleinere hoeveelheid feromonen achterlaten, of als een kleiner aantal mieren in dit geval feromonen achterlaat. Merk op dat we ook veronderstellen dat de snelheid waarmee mieren het spoor verliezen, afhangt van het aantal mieren op het spoor en niet van de sterkte van het spoor, wat een vereenvoudiging is. Wanneer de afstand tot de voedselbronnen verschillend is, zullen ook α, s en c afhankelijk zijn van de voedselbron. Analyse van het model [72] leert ons dat het grootste deel van de mieren voor de beste voedselbron zal kiezen. Dit wordt bevestigd door experimentele resultaten in [72] en wordt hoofdzakelijk toegeschreven aan het niet lineair zijn van differentiaalvergelijking (1.5). Wanneer twee identieke voedselbronnen worden aangeboden, wordt gekozen voor één ervan. In tegenstelling tot bijvoorbeeld honingbijen, worden de mieren niet evenwichtig verdeeld over de twee bronnen. Een gevolg hiervan is dat wanneer een betere voedselbron ontdekt wordt door de mieren nadat een sterk feromonenspoor naar een andere voedselbron reeds geconstrueerd werd, de mieren de inferieure voedselbron zullen blijven bezoeken [72]. In deze zin is het gedrag van mieren slechts suboptimaal. Een laatste bemerking is dat het totale aantal mieren dat in het geval van een enkele voedselbron het spoor zal volgen, onafhankelijk is van de sterkte van de bron. De sterkte van de bron zal enkel een invloed hebben op de snelheid waarmee het foerageren op gang komt [72].

1.5.3 Een fysisch model

In [33] worden enkele ideeën i.v.m. (nonequilibrium) thermodynamica toegepast op een simulatie van het foerageren bij mieren. In deze simulatie bevinden de mieren zich op een 2-dimensionaal rooster. Op het rooster is ook een nest aangebracht en een aantal voedselbronnen. Initieel bevinden alle mieren zich in het nest. Mieren die op zoek zijn naar voedsel laten nestferomonen achter. Hun beweging wordt gestuurd door de aanwezigheid van voedselferomonen. Mieren die voedsel aan het dragen zijn laten voedselferomonen achter en volgen nestferomonen. Het volgen van feromonensporen wordt op een probabilistische manier beschreven. In het begin komen de mieren geen feromonen tegen en zijn dus volledig onwetend over de beste richting die ze kunnen uitgaan vanaf het vakje waarop ze zich bevinden. Naarmate hogere feromonenconcentraties voorkomen, worden de mieren meer en meer beperkt in hun vrijheid. De mate waarin de mieren geordend gedrag vertonen wordt berekend a.d.h.v. de entropie. De volgende beweringen uit de thermodynamica zijn hier van toepassing [33]:

- In de initialisatiefase leidt een stijging van de (spatiale) entropie tot een vermindering van de onwetendheid en bewegingsvrijheid van de mieren.
- De structuur ontstaat als gevolg van positieve feedback: de vermindering van de entropie wordt veroorzaakt door, de beperkingen die aan het gedrag van de mieren wordt opgelegd. Anderzijds worden deze beperkingen zelf veroorzaakt door de vermindering van de entropie.
- De aanwezigheid van een voedselbron kan gezien worden als de externe beperking die noodzakelijk is om de structuur te behouden, en die het systeem ver van een evenwichtstoestand verwijderd houdt.

Dit houdt verband met de manier waarop het systeem leert van de omgeving. Het leren kan hierbij gemeten worden a.d.h.v. de beperkingen die opgelegd worden aan (of equivalent de bewegingsvrijheid van) de mieren. In het algemeen kan een agentgebaseerd systeem leren door wijzigingen in het interactiepatroon tussen de agenten, door wijzigingen in de interne regels waarmee de agenten gestuurd worden en door wijzigingen in de potentiële informatie die in de omgeving ligt opgeslagen (vb. feromonensporen) [33].

1.6 Organiseren van kerkhoven

1.6.1 Het basismodel

We stelden reeds in paragraaf 1.4 dat mieren hun dode soortgenoten herkennen door de aanwezigheid van oliezuur. Wat daarna met de kadavers gebeurt is sterk afhankelijk van soort tot soort. Sommige miersoorten eten de kadavers op. Meestal worden de kadavers echter door werkmieren uit het nest verwijderd. Deze verlaten het nest in een willekeurige richting en laten het kadaver achter op een onvoorspelbare afstand van het nest. Nog andere soorten hebben hiervoor speciale afvalkamers (refuse chambers) in het nest [41]. Chrétien stelde experimenteel vast bij de miersoort Lasius Niger dat werkmieren de kadavers van dode soortgenoten die verspreid liggen over een gebied groeperen in hopen. Deneubourg et al. stelden het volgende eenvoudig model op dat het achterliggende mechanisme verklaart [23]. We veronderstellen dat mieren rondwandelen in een omgeving waar dode mieren aanwezig zijn. Een mier die geen kadaver aan het dragen is, zal een kadaver opnemen in zijn omgeving met een probabiliteit die gegeven wordt door

$$P_p = \left(\frac{k_1}{k_1 + f}\right)^2 \tag{1.6}$$

Hierbij is k_1 een positieve reële constante en f de fractie van de waargenomen omgeving waar dode mieren liggen. Hoe meer dode mieren er in de buurt van de mier liggen, hoe kleiner dus de kans dat de mier een kadaver zal opnemen. Een mier die een kadaver aan het dragen is, zal dit neerleggen met een probabiliteit die gegeven wordt door:

$$P_d = \left(\frac{f}{k_2 + f}\right)^2 \tag{1.7}$$

Hierbij is k_2 een positieve reële constante. Hoe meer dode mieren aanwezig zijn in de omgeving, hoe groter dus de kans dat het kadaver wordt neergelegd. Deneubourg veronderstelde dat de fractie f werd berekend met behulp van het kortetermijngeheugen van de mieren. Noem T het aantal tijdseenheden dat de mier kan onthouden en noem N het aantal dode mieren dat de mier tegengekomen is in de laatste T tijdseenheden. Wanneer we nu veronderstellen dat de mier gedurende elke tijdseenheid ten hoogste 1 kadaver kan tegenkomen, kunnen we $f = N/T$ stellen. Deze manier om f te berekenen vertaalt zich gemakkelijk naar een robotimplementatie. Nochtans zullen echte mieren zich waarschijnlijk baseren op chemische en fysische hints en zal in algoritmen die gebaseerd zijn op dit principe een meer directe evaluatie mogelijk zijn [11]. Sommige mieren organizeren hun nest zodanig dat er een stapel met eieren, een stapel met larven en een stapel met coconnen ontstaat [40]. Bij sommige soorten zijn deze stapels naast elkaar gelegen, bij andere soorten bevinden deze stapels zich in verschillende delen van het nest. Dit sorteren gebeurt bovendien niet steeds in stapels, zo

worden bijvoorbeeld soms concentrische cirkels gebruikt waarbij de afstand van het centrum correspondeert met de ontwikkelingsfase van de toekomstige mieren die er gelegen zijn [40]. Deneubourg gebruikte hetzelfde model als hetgeen hierboven beschreven staat om ook dit sorteren te verklaren. Hierbij worden bij de berekening van f enkel de items van dezelfde soort gebruikt. Hoewel de biologische relevantie van dit model voor het sorteren sterk in twijfel wordt getrokken [11], heeft het geleid tot het invloedrijke algoritme van Lumer en Faieta [56] voor het clusteren van data waar we in het volgende hoofdstuk dieper op ingaan. We zullen verder over aggregatie spreken wanneer alle objecten gelijk zijn en over sorteren in het andere geval. De term clusteren zullen we voor beide gevallen gebruiken.

In het volgende hoofdstuk zullen we zien hoe we ons op dit gedrag kunnen inspireren om data te clusteren. Het is hiervoor van fundamenteel belang de principes die aan de basis liggen van dit gedrag ten volle te begrijpen. In tegenstelling tot de modellen die we hiervoor besproken hebben, lijkt het hier niet om een vorm van zwermintelligentie te gaan. Een individuele mier zou net zo goed de kadavers kunnen clusteren als een groep mieren. Dit wordt bevestigd door experimentele resultaten in [58]. Er is echter wel sprake van een vorm van stigmergie in die zin dat het neerleggen van een kadaver op een bepaalde plaats, het toekomstig neerleggen van kadavers in de buurt van die plaats bevordert. We bespreken nu twee varianten op het model van Deneubourg. Deze alternatieven hebben enkel tot doel beter te begrijpen waarom het model van Deneubourg werkt en proberen niet te verklaren hoe echte mieren hun kerkhoven construeren.

1.6.2 Complexiteit-zoekende mieren

In [34] wordt gesteld dat het achterliggende mechanisme een combinatie is van twee processen. Een eerste proces vermindert de complexiteit op lokaal niveau, een tweede zorgt ervoor dat een vermindering in lokale complexiteit ook een vermindering van de complexiteit op grote schaal tot gevolg heeft. De globale entropie wordt met andere woorden verminderd door het verminderen van de lokale complexiteit door de mieren. De berekening van de parameter f in (1.6) en (1.7) gebeurt zoals we reeds aanhaalden aan de hand van een kortetermijngeheugen. Hierdoor wordt de lokale complexiteit op een impliciete manier in rekening gebracht. In [34] wordt deze (lokale) complexiteit expliciet gemaakt. Complexiteit is gerelateerd met de dichtheid van de objecten. Wanneer de dichtheid hoog of laag is, zal de complexiteit laag zijn. Voor tussenliggende waarden echter zal de complexiteit hoog zijn. Als lokale omgeving van een vakje, beschouwen we het 3×3 rooster dat gecentreerd is rond dit vakje. In dit 3×3 rooster komen nu 12 overgangen voor tussen twee vakjes (6 verticaal en 6 horizontaal). De lokale complexiteit wordt dan gedefinieerd als het aantal dergelijke overgangen tussen 2 vakjes waarbij de 2 vakjes zich in een verschillende toestand bevinden. Met toestand wordt het al dan niet bevatten van een kadaver bedoeld. Als alle 9 vakjes ingenomen worden door een dode mier of als alle 9 vakjes vrij zijn, zal de complexiteit 0 zijn. Wanneer de 9 vakjes een dambordpatroon vertonen, zal de complexiteit zijn maximale waarde 12 bereiken. Het model van Deneubourg wordt nu als volgt aangepast: de mier zal in dezelfde richting verder blijven bewegen en geen kadavers opnemen of neerleggen met een probabiliteit die gegeven wordt door [34]

$$P = \frac{12 - (C + T)}{12}$$

Hierbij is C de lokale complexiteit en $T \in \{0, 1, \ldots, 12\}$ een drempelwaarde. In het andere geval zal willekeurig een nieuwe richting gekozen worden en zal met de probabiliteiten van het

oorspronkelijke model een dode mier opgenomen of neergelegd worden. Hierdoor wordt de reductie van de lokale complexiteit door de mieren expliciet gemaakt. De mieren uit dit model worden complexiteit-zoekende mieren genoemd omdat ze weinig tijd spenderen in gebieden waar de complexiteit laag is. Uit experimenten in [34] blijkt dat de entropie op kleine schaal snel verkleint en na korte tijd tot een evenwichtstoestand komt. De entropie op grote schaal verkleint gradueel. De overgang van kleine tot grote clusters wordt mogelijk gemaakt door stochastische fluctuaties [34]. Beschouwen we bijvoorbeeld het geval waarbij nog slechts twee clusters van gelijke grootte overblijven. Stochastische fluctuaties zullen er uiteindelijk toe leiden dat de grootte van één van de twee clusters de andere domineert. De mieren zullen er dan voor zorgen dat dit verschil in grootte vervolgens versterkt wordt. Hierdoor blijft er op lange termijn slechts 1 cluster over.

1.6.3 Het minimale model

We stelden reeds dat het clusteren van dode lichamen niet noodzakelijk steunt op zwermintelligentie. In [58] wordt nog een stap verder gegaan. Ook de intelligentie van de mieren die hen in staat stelt de lokale densiteit van de kadavers waar te nemen en te interpreteren, blijkt overbodig. Enkel de volgende regels worden gebruikt [58]:

- Als een vrije mier een kadaver ziet in één van de naburige vakjes (alle acht richtingen worden beschouwd), wordt dit kadaver met probabiliteit 1 opgenomen. Wanneer er verschillende mogelijkheden zijn, wordt een willekeurige keuze gemaakt.
- Als een beladen mier een kadaver ziet in één van de naburige vakjes en als de mier ten minste één stap verwijderd is van de plaats waar hij het kadaver heeft opgenomen, wordt het kadaver met probabiliteit 1 op een van de naburige (vrije) vakjes neergelegd.

In dit model kan een mier niet langer op een vakje lopen waar een kadaver ligt. Bovendien worden niet langer volledig willekeurige bewegingen genomen. In plaats hiervan wordt een random waarde gekozen uit $\{1, 2, \ldots, l\}$ voor een zekere constante l. De mier beweegt dan l stappen in dezelfde richting en kiest vervolgens een nieuwe willekeurige richting. Deze wijziging werd hoofdzakelijk doorgevoerd om efficiëntieredenen. Het merkwaardige resultaat is dat ook op deze manier uiteindelijk 1 enkele cluster gevormd wordt. De snelheid is ten minste een factor 10 lager dan in het model van Deneubourg en er ontstaat een losse cluster waarin vrije en bezette vakjes door elkaar voorkomen. Desalniettemin is duidelijk dat de intelligentie waarmee de mieren in het basismodel begunstigd zijn, geen nodige voorwaarde is. Aangezien een kadaver steeds wordt neergelegd naast een ander kadaver, kan het totale aantal clusters nooit stijgen. Nochtans heeft elke bestaande cluster een zekere kans om te verdwijnen. Deze kans is groter voor kleine clusters. Bijgevolg zullen eerst de kleine clusters verdwijnen. Na verloop van langere tijd zullen ook bepaalde grote clusters verdwijnen of samensmelten, waardoor we uiteindelijk slechts een enkele cluster overhouden.

1.6.4 Robotimplementaties

Met dezelfde doelstellingen als het minimale model werden ook robotimplementaties ontwikkeld [4, 40]. De motivatie voor het gebruik van robots i.p.v. simulaties is de volgende. Stigmergie werkt steeds dankzij fysische eigenschappen van de omgeving. Aangezien in computersimulaties noodzakelijk een drastische vereenvoudiging optreedt van deze eigenschappen,

zijn robotimplementaties in veel opzichten geschikter om stigmergische processen te bestuderen [40]. Het traditionele roboticaparadigma bestaat uit het waarnemen van de omgeving, het bouwen van een wereldmodel op basis van de gedetecteerde kenmerken, redeneren over de uit te voeren taak en het wereldmodel om een sequentie van acties te bepalen en tenslotte het één voor één uitvoeren van de gevonden acties waarbij het wereldmodel aangepast wordt en de acties herpland worden waar nodig [4]. In de praktijk is het gebruik van dergelijke robots in een ongestructureerde dynamische omgeving niet haalbaar [4]. In tegenstelling hiermee bestaat een gedragsgebaseerde robot uit een beperkt aantal eenvoudige modules die elk een klein aantal aspecten van de omgeving kunnen waarnemen. Preferenties worden hardgecodeerd en het gedrag met de hoogste preferentie krijgt de controle over de volledige robot [4]. Het is duidelijk dat dergelijke gedragsgebaseerde robots te verkiezen zijn boven traditionele robots bij het onderzoeken van stigmergische principes voor robots.

In [4] wordt een experiment beschreven waarbij een aantal robots zich in een afgebakende ruimte bevinden. Elke robot is voorzien van een C-vormige grijper waarmee de robots de schijven die zich in de omgeving bevinden kunnen vooruitduwen. De robots zijn zo geprogrammeerd dat ze steeds in een rechte lijn bewegen. Wanneer ze echter een hindernis of de rand naderen, stoppen ze en kiezen ze willekeurig een nieuwe richting. De robot weet ook hoeveel schijven hij aan het duwen is. Op het moment dat drie (of meer) schijven geduwd worden, worden deze achtergelaten door de robot (de robot beweegt achteruit en kiest willekeurig een nieuwe richting). Zoals verwacht worden de schijven snel in kleine groepjes verzameld. Na verloop van een voldoende lange tijd (bijvoorbeeld 10 uur) echter liggen alle schijven bij elkaar in 1 cluster. Een verklaring voor het feit dat een dergelijk eenvoudig gedrag het groeperen van alle schijven in een enkele cluster tot gevolg heeft, gebeurt aan de hand van stigmergie. Beschouwen we voor de eenvoud alle clusters als cirkels. Een robot kan enkel een schijf wegnemen van een cluster door deze zijdelings te benaderen. Een frontale benadering zal resulteren in het achterlaten van de schijven die de robot aan het duwen was. De kans dat een cluster zijdelings of frontaal benaderd wordt, is afhankelijk van de grootte en vorm van de cluster. Door objecten toe te voegen wordt de cluster groter en wordt de kans om de cluster voldoende zijdelings (resp. frontaal) te raken kleiner (resp. groter). Het wegnemen (resp. toevoegen) van schijven aan een cluster stimuleert dus het toekomstig wegnemen (resp. toevoegen) van schijven. Merk op dat deze techniek robuust is m.b.t. het falen van een enkele robot (deze robot wordt dan door de andere robots gewoon als hindernis gezien).

In [40] wordt een verklaring gezocht voor het sorteren in concentrische cirkels met behulp van een robotimplementatie. Er wordt gebruik gemaakt van een robot met een grijper die één schijf kan verplaatsen. In de ruimte zijn nu twee verschillende soorten schijven aanwezig: een soort die volledig wit is en een soort waarbij een zwarte band op een witte achtergrond aanwezig is. De robots kunnen uiteraard het verschil tussen deze soorten waarnemen. De bewegingen van de robot zijn dezelfde als in het vorige experiment. Wanneer de robot een schijf met een zwarte rand vast heeft, wordt deze achtergelaten zodra de robot op een andere schijf botst. Als de robot daarentegen een volledig witte schijf vast heeft, zal deze na het detecteren van een andere schijf, een korte afstand achteruit gaan vooraleer de schijf los te laten. Het merkwaardige resultaat is dat een hechte cluster van schijven met zwarte rand ontstaat waarrond de witte schijven gelegen zijn. Merk op dat het clusteren hier ontstaan is zonder enige vorm van waarneming van de lokale dichtheid van de aanwezige objecten. Wanneer objecten op stapels gelegd zouden worden, zou dit echter niet langer mogelijk zijn [40]. Het achteruit laten gaan van de robots om de witte en de gestreepte schijven te sorteren is eveneens gebaseerd op het gedrag van mieren, zij het niet bij gedrag dat geobserveerd werd

bij het sorteren van het broed [40].

1.6.5 Analyse van het globale gedrag

Om inzicht te krijgen in het globale gedrag dat hieruit ontstaat, werd in [75] een model opgesteld dat gebaseerd is op reactie-diffusie vergelijkingen. Deze reactie-diffusie vergelijkingen werden in de jaren '50 van de vorige eeuw ingevoerd door Alan Turing om het ontstaan van bepaalde patronen die het gevolg zijn van chemische reacties of fysische processen te verklaren. De bekendste toepassing betreft het verklaren van de patronen op de huid van giraffen, zebra's , ... De algemene vorm wordt gegeven door [11]

$$\frac{\partial A}{\partial t} = F(A,B) + D_A \nabla^2 A$$
$$\frac{\partial B}{\partial t} = G(A,B) + D_B \nabla^2 B$$

Met A en B afhankelijk van zowel de tijd t als de plaats r. Hierbij is $\nabla^2 A$ de Laplaciaan van A, gegeven door $\nabla^2 A = \frac{\partial^2 A}{\partial r^2} + \frac{\partial^2 A}{\partial t^2}$; D_A en D_B worden de diffusieconstanten genoemd. In de context van chemische reacties stellen A en B de concentraties van 2 stoffen voor. Daarbij is het meestal zo dat de productie van A, de productie van B versterkt of tegenwerkt. Er kan worden aangetoond dat dan onder bepaalde voorwaarden voor de diffusiecoëfficiënten patronen ontstaan die Turing-patronen genoemd worden. We beschrijven nu een model voor het clusteren van dode mieren op basis van reactie-diffusie vergelijkingen [75]. Noem $a(x,t)$ de dichtheid van mieren die een kadaver aan het dragen zijn op plaats x en tijdstip t en $c(x,t)$ de dichtheid van kadavers. De dichtheid van de mieren die geen kadaver dragen, wordt aangeduid met ρ en wordt verondersteld uniform te zijn en constant in de tijd. We kunnen het gedrag van de mieren dan beschrijven met behulp van volgende reactie-diffusie vergelijkingen [75]:

$$\frac{\partial c}{\partial t} = \Omega(c,a) \tag{1.8}$$
$$\frac{\partial a}{\partial t} = -\Omega(c,a) + D\frac{\partial^2 a}{\partial x^2} \tag{1.9}$$

De diffusiecoëfficiënt van (1.8) is 0 aangezien dode mieren op exact dezelfde plaats blijven zolang ze niet opgenomen worden door een mier. In (1.9) is de diffusiecoëfficiënt een constante D die experimenteel moet worden vastgesteld. Merk op dat $\frac{\partial^2 a}{\partial t^2} = 0$ geldt. Hierbij is

$$\Omega(c,a) = v\left(k_d a + \frac{\alpha_1 a \phi_c}{\alpha_2 + \phi_c} - \frac{\alpha_3 \rho c}{\alpha_4 + \phi_c}\right) \tag{1.10}$$

met v, k_d, α_1, α_2, α_3 en α_4 reële constanten en

$$\phi_c = \frac{1}{2\Delta}\int_{x-\Delta}^{x+\Delta} c(z)dz$$

Waarbij Δ de straal is waarbinnen mieren kadavers kunnen waarnemen. De constante v stelt de snelheid van de mieren voor, die voor de eenvoud gelijk voor alle mieren wordt verondersteld. Mieren leggen een kadaver dat ze aan het dragen zijn bij voorkeur neer in een

omgeving waar de dichtheid van de kadavers groot is. De tweede term correspondeert hiermee en is een stijgende functie van ϕ_c. Uiteraard is deze term ook afhankelijk van de dichtheid van de kadaverdragende mieren. De eerste term correspondeert met het feit dat mieren met een kleine probabiliteit de kadavers die ze aan het dragen zijn ergens willekeurig zullen achterlaten. Het opnemen van kadavers is een dalende functie van ϕ_c en wordt beschreven door de derde term. Deze term is bovendien afhankelijk van ρ en c aangezien om een kadaver op te nemen er zowel een vrije mier als een kadaver moet aanwezig zijn. De parameters die voorkomen in dit model moeten alle experimenteel worden vastgesteld. Gewapend met deze parameterwaarden kan men dan aan de hand van dit model de eigenschappen van het ontstaan van de kerkhoven nabootsen [75]. Bovendien kan men voorspellen hoe het wijzigen van bepaalde parameters het resultaat zal beïnvloeden. Zo wordt in [75] bijvoorbeeld aan de hand van een stabiliteitsanalyse voorspeld en experimenteel bevestigd:

- Het verdubbelen van de dichtheid van de kadavers leidt onder bepaalde voorwaarden tot een verdubbeling van het aantal hopen. Deze situatie kan echter veranderen na langere tijd.
- Wanneer we de omgeving waarin de mieren zich kunnen begeven verdubbelen en de dichtheid van de kadavers constant houden zal dit leiden tot een verdubbeling van het aantal hopen.
- Er bestaat een kritische dichtheid waaronder geen aggregatie optreedt. Bovendien kan de waarde van deze kritische dichtheid theoretisch bepaald worden.

Hoofdstuk 2

Mieralgoritmen voor het clusteren van data

2.1 Inleiding

In de biologie zijn vele systemen te vinden waarbij eigenschappen op het globale niveau niet onmiddellijk te herleiden zijn naar eigenschappen van lokale processen. Zoals we in het vorige hoofdstuk gezien hebben zijn sociale insecten zoals mieren en bijen hier goede voorbeelden van. Ook immuunsystemen vallen onder deze noemer [56]. Vooral de opkomst van snellere computers heeft ertoe geleid dat eigenschappen van populaties experimenteel konden worden vastgesteld. Deze inzichten hebben op hun beurt tot inspiratie geleid voor het ontwerpen van gedistribueerde systemen en algoritmen. We bespreken in dit hoofdstuk een aantal clusteringsalgoritmen die op een of andere manier geïnspireerd zijn op het gedrag van mieren. In het bijzonder zijn de meeste hiervan geïnspireerd op de manier waarop mieren hun kerkhoven organiseren.

Steeds kan gebruik gemaakt worden van het parallellisme dat intrinsiek aanwezig is en van het ontbreken van centrale controle [56]. De meest logische aanpak om van dit parallellisme gebruik te maken zou zijn om aan elke processor die we ter beschikking hebben een aantal mieren toe te wijzen. Deze aanpak is echter niet bruikbaar wegens de grote communicatie-overhead die hierbij ontstaat. Zelfs bij een model waarbij de processoren over een gemeenschappelijk geheugen beschikken zal de overhead die ontstaat door het gebruik van synchronisatieprimitieven waarschijnlijk te hoog zijn. Een oplossing die wel succesvol blijkt te zijn, is het rooster waarop de mieren zich bevinden op te splitsen en aan elke processor een deel van het rooster toe te wijzen [1]. Communicatie tussen de processoren is dan enkel nog nodig voor de randen van elk gebied.

De algoritmen die we bespreken zijn nagenoeg allemaal stochastisch van aard. Zoals we in paragraaf 1.5.3 besproken hebben houdt de vrijheid en dus de willekeur waarmee agenten handelen verband met de mate waarin de agenten de structuur van de omgeving geleerd hebben. In [34] wordt een gelijkaardige filosofische kijk op het algoritmisch modelleren van het gedrag van dieren voorgesteld. We beschouwen hiertoe elke omgeving als de combinatie van voorspelbare en onvoorspelbare aspecten. Een organisme dat zich perfect heeft aangepast aan de omgeving is een organisme dat stereotiep gedrag vertoont bij uitdagingen die het gevolg zijn van voorspelbare aspecten van de omgeving en terugvalt op meer experimentele en meer risicovolle strategieën bij confrontatie met onvoorspelbare aspecten van de omgeving [34]. Om

de relatieve mate van aangepastheid aan de omgeving te meten bij (artificiële) agenten kunnen we gebruik maken van de hoeveelheid willekeur die nodig is om een bepaalde taak uit te voeren. Deze hoeveelheid willekeur wordt weerspiegeld in het aantal randomwaarden dat nodig is om de taak uit te voeren, en in de grootte van de verzameling waaruit deze randomwaarden gekozen worden. We kunnen de hoeveelheid willekeur dus kwantitatief bepalen als het aantal bits dat willekeurig gekozen wordt bij de uitvoering van de taak. Hoe minder randombits nodig zijn, hoe meer we kunnen stellen dat de agent aangepast is aan de omgeving. Hoewel het mogelijk is deterministische clusteringsalgoritmen op te stellen die gebruik maken van globale evaluatiecriteria, zullen mieren met enkel de mogelijkheid om de lokale omgeving waar te nemen bepaalde keuzes moeten maken op basis van enkel een schatting van de hiervoor noodzakelijke informatie. In [34] wordt geargumenteerd dat hierbij steeds een minimum aan willekeur noodzakelijk is. We bespreken nu eerst op welke manier het model van Deneubourg voor aggregatie uit het vorige hoofdstuk kan uitgebreid worden tot een clusteringsalgoritme voor data.

2.2 Van aggregatie naar visualisatie

2.2.1 Het basisalgoritme

Het clusteren van een bepaalde gegevensverzameling wordt meestal informeel gedefinieerd als het onderverdelen van de objecten in de gegevensverzameling in groepen, die clusters genoemd worden, waarbij het zo is dat twee objecten binnen een zelfde cluster sterker op elkaar gelijken dan twee objecten uit verschillende clusters. Lumer en Faieta beschreven in [56] volgend algoritme dat een uitbreiding is van het model van Deneubourg voor het clusteren van niet-homogene objecten. Mieren wandelen willekeurig rond op een tweedimensionaal rooster waarop de te clusteren objecten willekeurig verspreid liggen. Dit wordt geïllustreerd in figuur 2.1[1]. Dikwijls (bijvoorbeeld in [39], [44] en [59]) wordt hierbij verondersteld dat het rooster toroïdaal is, de linker- en rechterrand en de onder- en bovenrand zijn met andere woorden met elkaar verbonden. Wanneer we dit niet veronderstellen zullen vakjes aan de randen van het rooster meer bezocht worden dan vakjes in het centrum, wat een ongewenste eigenschap is. Hierbij is het zo dat op elk vakje van het rooster hoogstens 1 object aanwezig is. De grootte van het rooster wordt zodanig gekozen dat het aantal vakjes het aantal objecten ruimschoots overschrijdt. Wanneer er op het rooster onvoldoende vrije vakjes aanwezig zijn, zullen er, wegens plaatsgebrek, geen clusters van gelijkaardige objecten kunnen ontstaan. Bovendien wordt ook het aantal mieren door het aantal objecten ruimschoots overschreden. Wanneer dit niet zo zou zijn, zou het algoritme gehinderd worden door het feit dat bijna alle objecten door de mieren gedragen worden en er dus onvoldoende objecten op het rooster overblijven om clusters te laten ontstaan.

Bij het clusteren maken we gebruik van discrete tijdstappen. In elke tijdstap wordt er willekeurig 1 mier uitgekozen. Indien deze mier nog geen object aan het dragen was en indien er een object aanwezig is op het vakje waarop de mier zich bevindt, kan deze dit object opnemen. In het geval dat de mier reeds een object aan het dragen was en zich op een vrij vakje bevindt, kan deze beslissen het object neer te leggen. Ten slotte wordt volledig willekeurig 1 van de 4 mogelijke richtingen gekozen (de diagonalen werden door Lumer en Faieta niet

[1]De implementatie waarvan gebruik gemaakt werd om deze figuur te bekomen, is een kleine variant op het algoritme van Lumer en Faieta waarin gebruik gemaakt wordt van vaagregels bij het evalueren van de lokale densiteit. We komen verder nog uitgebreid terug op het gebruik van vaagregels.

beschouwd) en wandelt de mier 1 vakje ver in de gekozen richting. De probabiliteit dat een object i wordt opgenomen door een vrije mier wordt gegeven door

$$P_{pick}(i) = \left(\frac{k_p}{k_p + f(i)} \right)^2 \tag{2.1}$$

Hierbij is k_p een strikt positieve, reële constante en $f(i)$ een schatting van de lokale dichtheid van het aantal objecten en hun gelijkenis met het object i. De kans dat het object i wordt neergelegd, wordt gegeven door

$$P_{drop}(i) = \begin{cases} 2f(i) & \text{als } f(i) < k_d \\ 1 & \text{anders} \end{cases} \tag{2.2}$$

Met k_d een constante in $[0, \frac{1}{2}]$. De functie f wordt voor een gegeven object i gedefinieerd als:

$$f(i) = \max \left(\frac{1}{r^2} \sum_{j \in N(i)} \left(1 - \frac{d(i,j)}{\alpha} \right), 0 \right) \tag{2.3}$$

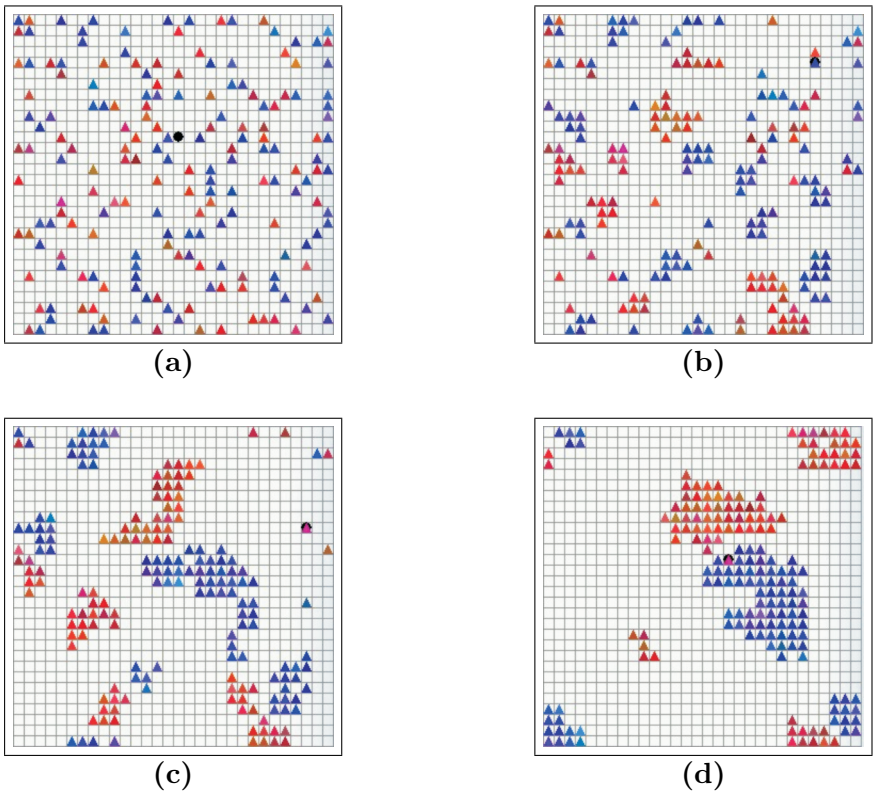
Hierbij is $N(i)$ de verzameling van alle objecten die zich in de omgeving van i bevinden. De omgeving wordt hierbij gedefinieerd als het $r \times r$ rooster dat gecentreerd is rond het vakje waarop i zich bevindt. De waarde $d(i,j)$ stelt de mate van dissimilariteit voor tussen i en j. Meestal wordt hiervoor de Euclidische afstand gebruikt. Deze afstand wordt herschaald m.b.v. de constante schaalfactor α. Merk op dat we niet eisen dat $\alpha \geq \max_{i,j} d(i,j)$ waardoor sommige termen van de som negatief kunnen zijn. Sterk verschillende objecten in de omgeving kunnen met andere woorden een negatieve invloed uitoefenen. De factor $\frac{1}{r^2}$ heeft als gevolg dat lege vakjes gepenaliseerd worden. De functie f evalueert dus zowel de similariteit met een gegeven object als de densiteit van de objecten in de omgeving.

Multidimensionale schaling (multi-dimensional scaling) wordt gedefinieerd als het representeren van een gegevensverzameling, beschreven als een verzameling vectoren in een n-dimensinale vectorruimte, op een 2- of 3-dimensionale kaart. Hierbij is n een, typisch groot, natuurlijk getal. Het is hierbij gewenst dat de afbeelding die deze voorstelling realiseert zo goed mogelijk afstandsbewarend is, m.a.w. als een object y in de n-dimensionale vectorruimte dichter bij een object x gelegen is dan een object z, wensen we dat dit eveneens geldt voor hun representaties op de kaart. Het is duidelijk dat dit niet altijd mogelijk is en we ons tevreden moeten stellen met een benadering.

Het resultaat van het algoritme van Lumer en Faieta lijkt enerzijds sterk op een clustering van de data in die zin dat de similariteit tussen de objecten in een zelfde cluster groot is. De similariteit tussen objecten van verschillende clusters is echter niet noodzakelijk klein; het aantal clusters wordt dus overschat. Dit wordt geïllustreerd in figuur 2.1 (d). Anderzijds vertoont het algoritme ook sterke gelijkenissen met een multidimensionale schaling van de data wat betreft de intraclusterstructuur. De onderlinge posities van de clusters zijn echter zo goed als willekeurig.

2.2.2 Uitbreidingen

Om tegemoet te komen aan de tekortkomingen van het basisalgoritme suggereerden Lumer en Faieta zelf reeds enkele verbeteringen [56]:



Figuur 2.1: Variant op het algoritme van Lumer en Faieta na: (a) 0 stappen, (b) 10^4 stappen, (c) 10^5 stappen, (d) 10^6 stappen

Heterogene populaties We maken gebruik van mieren met verschillende snelheden. De snelheid van een mier is hier het aantal vakjes dat de mier in 1 stap vooruitspringt. Verschillende mieren kunnen een verschillende snelheid hebben die kan variëren tussen 1 en V_{max}. De snelheid van een bepaalde mier blijft echter constant. Bovendien wordt de snelheid v van een mier gekoppeld aan zijn tolerantie m.b.t. de similariteit van de objecten. Dit wordt uitgedrukt door de volgende definitie voor f te gebruiken:

$$f(i) = \max\left(\frac{1}{d^2}\sum_j\left(1 - \frac{d(i,j)}{\alpha + \alpha(v-1)/V_{max}}\right), 0\right)$$

Een grote verbetering m.b.t. het resulterende aantal clusters werd gerapporteerd. De snelheid van elke mier werd hiervoor uniform gekozen uit de verzameling $\{1, 2, \ldots, 6\}$. De reden voor de verbetering is dat de snelle mieren een ruwe classificatie maken, die vervolgens verfijnd wordt door de trage mieren.

Kortetermijngeheugen De mieren onthouden de laatste m objecten die ze neergelegd hebben, samen met de plaats waar ze die neergelegd hebben. Hierbij is m een natuurlijk getal (in tegenstelling tot echte mieren, clusteren deze artificiële mieren dus nooit stukken van objecten). Als ze een nieuw object opnemen, wordt dit object vergeleken met de objecten in het geheugen. De mier gaat vervolgens automatisch naar de locatie behorende bij het meest gelijkaardige object in zijn geheugen.

Gedragsschakelaars (Behavioral switches) Naarmate alle objecten meer en meer in een aanvaardbare omgeving terechtkomen, wordt het minder en minder waarschijnlijk dat ze nog verplaatst zullen worden naar een nieuwe locatie. De mieren zullen dus na verloop van tijd minder en minder objecten verplaatsen. Daarom zorgen we ervoor dat elke mier die meer dan een voorgedefinieerd aantal iteraties niets meer gedaan heeft, overgaat tot het vernietigen van een cluster.

In [37] wordt dit algoritme toegepast voor het visualiseren van de resultaten van een internet zoekmachine. We zullen verder nog uitvoerig terugkomen op het bepalen van de similariteit tussen documenten. Voorlopig volstaat het te weten dat de documenten vaak beschreven worden als een vector in een n-dimensionale vectorruimte. Hierbij is n het totale aantal verschillende termen dat voorkomt in de documentenverzameling. Om deze documenten te visualiseren moet deze n-dimensionale documentruimte nu worden voorgesteld in een 2-dimensionale visualisatieruimte, we hebben m.a.w. nood aan een multidimensionale schaling van de documenten. Zelfs na de voorgestelde uitbreidingen vertoont het model van Lumer en Faieta nog belangrijke beperkingen met betrekking tot dit probleem [37]. De afstanden tussen de individuele clusters zijn, zoals we reeds aanhaalden, niet representatief (geen afstandsbewarende inbedding). Bovendien kan de waarde van de parameter α, die een sterke invloed heeft op de goede werking van het algoritme, niet op voorhand bepaald worden (deze is afhankelijk van de data). Ten slotte vermelden we ook nog de trage convergentie, m.a.w. er is een groot aantal iteraties noodzakelijk om een aanvaardbare oplossing te bekomen. Handl en Meyer [37] stelden hiertoe de volgende wijzigingen voor:

Adaptieve schaling De parameter α krijgt een initiele waarde (vb. 0.1) en wordt telkens verhoogd met 0.01 als 250 stappen voorbijgegaan zijn waarin weinig verplaatsingen voorkwamen.

Stagnatiecontrole Elementen die van alle andere verschillen, blokkeren de mieren aangezien er geen geschikte plaats bestaat om ze achter te laten. Daarom wordt een teller bijgehouden die er voor zorgt dat de mieren na 100 mislukte pogingen hun object toch achterlaten.

Gretige mieren Wanneer we een lijst bijhouden van alle objecten die zich op het rooster bevinden, kunnen we de tijd die een mier spendeert aan het zoeken naar een geschikt object om op te nemen, volledig elimineren. In plaats van doelloos rond te dwalen kiest een mier die een object heeft achtergelaten, onmiddellijk een willekeurig object uit deze lijst. De mier springt vervolgens naar dat object en raapt het op met een probabiliteit die gegeven wordt door (2.1). In het geval er beslist wordt het object niet op te nemen, wordt er willekeurig een nieuw object uit de lijst gekozen.

Initialisatie Hoewel goede klassieke methoden voor multidimensionale schaling een hoge computationele kost met zich meebrengen, bestaan er enkele eenvoudige methoden die bruikbaar zijn als een eerste benadering. Om een hogere interclustercorrelatie te bekomen worden de objecten initieel op het rooster geplaatst m.b.v. een dergelijke eenvoudige benadering.

In [36] worden nog een aantal bijkomende wijzigingen voorgesteld. Zo wordt het gebruik van kortetermijngeheugen uitgebreid met een vooruitkijkmechanisme waardoor de omgeving van alle documenten in het geheugen geëvalueerd wordt. Bovendien wordt er een grotere

lokale omgeving gebruikt waarbij de evaluatie van (2.3) nu een gewogen som wordt met als gewichten de mate waarin de corresponderende objecten nog in de lokale omgeving gelegen zijn. Bij de evaluatie van (2.3) wordt de densiteit buiten beschouwing gelaten, er wordt een alternatief voorgesteld voor de probabiliteiten (2.1) en (2.2), ...

Het is duidelijk dat we heel wat van deze kleine ad hoc aanpassingen kunnen beschouwen. Deze doen echter in grote mate afbreuk aan de eenvoud en elegantie die het oorspronkelijke algoritme zo aantrekkelijk maakten. Bovendien brengen vele van deze aanpassingen een grote computationele kost met zich mee zonder dat het duidelijk is wat de impact van de aanpassing is. Het invoeren van nieuwe parameters zorgt er tevens voor dat het afstellen van de parameters verre van triviaal wordt, waarbij lang niet alle parameterwaarden tot een aanvaardbare oplossing leiden. Bij het oorspronkelijke algoritme spenderen mieren echter te veel tijd in zones waar geen objecten (meer) te vinden zijn, wat evenmin wenselijk is. Ramos et al. stellen als alternatief voor om hiervoor gebruik te maken van feromonensporen [67, 68]. Hiervoor wordt een uitbreiding gegeven van het model van Chialvo en Millonas dat we besproken hebben in sectie 1.4. De enige wijziging betreft de hoeveelheid feromonen die de mieren achterlaten per tijdseenheid. In plaats van hiervoor een constante η te beschouwen, wordt deze hoeveelheid afhankelijk gemaakt van het aantal objecten in de omgeving van de mier. Noem nu dit aantal objecten in de omgeving van een bepaalde mier Δ_h. Het aantal feromonen dat door deze mier wordt achtergelaten, wordt dan gegeven door $T = \eta + p\Delta_h$, waarbij zowel η als p constant zijn [67, 68]. Het gevolg is dat de hoeveelheid feromonen die achtergelaten wordt hoog zal zijn in gebieden waar veel objecten voorkomen. In gebieden waar geen objecten voorkomen zal de achtergelaten hoeveelheid hetzelfde zijn als in het model van Chialvo en Millonas. Aangezien dit model heel gevoelig is voor kleine perturbaties (zoals de aanwezigheid van een voedselbron of het initialiseren van de omgeving met een bepaalde feromonendistributie [14]) kunnen we verwachten dat de feromonensporen die ontstaan de mieren effectief naar plaatsen brengen waar objecten gelegen zijn. Dit wordt bevestigd door experimentele bevindingen in [67] en [68]. Aangezien deze feromonen ook verdampen, is dit mechanisme adaptief in die zin dat als een bepaalde cluster verdwijnt na een zekere tijd, ook de feromonensporen op deze plaats zullen verdwijnen. Door op deze manier gebruik te maken van feromonen zal de snelheid waarmee het clusteren gebeurt niet langer lineair afhankelijk zijn van het aantal mieren. Er treedt hier dus, in tegenstelling tot de andere clusteringsalgoritmen die we gezien hebben, wel een vorm van zwermintelligentie op.

Een laatste mogelijkheid, die in [8] voorgesteld werd, is de mieren niet langer te laten rondlopen, maar telkens een willekeurige positie te geven op het bord. Hierbij krijgt een mier die niets aan het dragen is de positie van een willekeurig object. In de volgende stap wordt beslist of het object wordt opgenomen of niet. Als het object niet wordt opgenomen, krijgt de mier de positie van een willekeurig ander object. Een mier die een object aan het dragen is krijgt de positie van een willekeurig vrij vakje. In de volgende stap wordt beslist of het object neergelegd wordt. Wanneer dit niet het geval is, krijgt de mier de positie van een willekeurig ander vrij vakje. We zullen hiertoe een lijst moeten bijhouden met de vakjes die door een object worden ingenomen en een lijst met de vrije vakjes.

2.3 Van visualisatie naar clustering

Het algoritme van Lumer en Faieta en zijn vele varianten zijn in het bijzonder geschikt voor het visualiseren van de data. Wanneer we echter enkel een clustering van de data willen

bekomen zonder visuele interpretatie kan heel wat gesleuteld worden aan de efficiëntie. Het beslissen om een object al dan niet neer te leggen op basis van de lokale densiteit is een gevolg van het feit dat het niet mogelijk is het object te vergelijken met de cluster waarin deze terechtkomt. Bovendien heeft deze methode als nadeel dat de elementen aan twee tegenovergestelde uitersten van een zelfde cluster totaal verschillend kunnen zijn. Ook het identificeren van de clusters is niet triviaal aangezien de randen van twee verschillende clusters elkaar kunnen raken [60]. Dit wordt geïllustreerd in figuur 2.1 (d).

Monmarché [59] stapte als eerste af van het idee om slechts 1 object per vakje toe te laten. Elk vakje waarop minstens 1 object ligt, wordt dan beschouwd als een afzonderlijke cluster. We zullen de verzameling objecten die op een vakje aanwezig is, een hoop noemen. Wanneer we nu moeten beslissen of we een object al dan niet neerleggen op een vakje kunnen we gebruik maken van informatie over de volledige hoop die op dat vakje ligt. Merk op dat we hiermee geen afbreuk doen aan het lokale karakter van mieralgoritmen, enkel informatie over 1 cluster wordt gebruikt. Dit staat in schril contrast met klassieke clusteringsalgoritmen die uitgaan van een globaal evaluatiecriterium. Voor een overzicht van enkele klassieke clusteringsalgoritmen verwijzen we naar appendix B. We zullen dit algoritme van Monmarché nu in detail bespreken. Zoals bij Lumer en Faieta worden de objecten initieel op een willekeurige manier op een vierkant rooster geplaatst. Wel werden volgende wijzigingen doorgevoerd:

- Het rooster is, net zoals bij de meeste varianten op het algoritme van Lumer en Faieta, toroïdaal.
- De grootte van het rooster wordt automatisch bepaald op basis van het aantal objecten. Zij N het aantal objecten (we veronderstellen verder steeds $N > 1$), de breedte en hoogte van het rooster worden dan gegeven door $L = \sqrt{2N}$.
- Een vakje van het rooster kan meerdere objecten bevatten en een cluster komt steeds overeen met de objecten die op 1 vakje gelegen zijn.

Aangezien we nu precies weten welke objecten er tot dezelfde cluster behoren, kunnen we een mier nu volledige hopen laten opnemen. Het aantal objecten dat een mier kan dragen noemen we de capaciteit van de mier; deze wordt voor een mier a gegeven door $c(a)$. In het bijzonder zullen de speciale gevallen $c(a) = 1$ en $c(a) = \infty$ beschouwd worden. We zullen veronderstellen dat we op één of andere manier het centrum van een hoop kunnen bepalen. Wanneer de objecten voorgesteld worden aan de hand van numerieke vectoren kunnen we als centrum de vector beschouwen met als waarde voor elk attribuut het gemiddelde van de waarden van dit attribuut bij alle objecten in de hoop. We spreken in dit geval over een centroïde. Wanneer geen numerieke vectoren gebruikt worden, kunnen we als centrum het object beschouwen waarvoor de gemiddelde dissimilariteit met de andere objecten van de hoop minimaal is. We spreken in dit geval van een medoïde. De gebruikte dissimilariteitsmaat is uiteraard afhankelijk van de toepassing. De richting waarin een mier beweegt is niet langer willekeurig, maar afhankelijk van de vorige richting. De mier zal met een probabiliteit 0.6 zijn huidige richting behouden en met een probabiliteit 0.4 links of rechts afslaan. Beschouwen we de volgende definities:

- De maximale dissimilariteit tussen de objecten van de objectenverzameling $O = \{x_1, \ldots, x_N\}$:
$$d^*(O) = \max_{i,j \in \{1,\ldots,N\}} d(x_i, x_j)$$

- De gemiddelde dissimilariteit tussen de objecten van O:

$$\overline{d}(O) = \frac{2}{N(N-1)} \sum_{i<j} d(x_i, x_j)$$

- De maximale dissimilariteit tussen de objecten van de hoop T_j en het centrum g_j van deze hoop:

$$d_g^*(T_j) = \max_{x_i \in T_j} d(x_i, g_j)$$

- De gemiddelde dissimilariteit tussen de objecten van de hoop T_j en het centrum g_j van deze hoop:

$$\overline{d_g}(T_j) = \frac{1}{|T_j|} \sum_{x_i \in T_j} d(x_i, g_j)$$

Een mier die niets aan het dragen is en zich op het vakje met een hoop T_j bevindt, zal objecten opnemen met een probabiliteit P_p, gegeven door

$$P_p(T_j) = \begin{cases} 1 & \text{als } |T_j| = 1 \\ \min\left(\left(\frac{\overline{d}_g(T_j)}{\overline{d}(O)}\right)^{k_1}, 1\right) & \text{als } |T_j| = 2 \\ 1 - 0.9\left(\frac{\overline{d}_g(T_j)+\varepsilon}{d_g^*(T_j)+\varepsilon}\right)^{k_1} & \text{anders} \end{cases}$$

Hierbij is ϵ een kleine, positieve waarde en k_1 een (strikt) positief natuurlijk getal. Het aantal objecten dat wordt opgenomen is gelijk aan de capaciteit van de mier. Deze objecten worden gekozen door achtereenvolgens het object weg te nemen waarvoor de dissimilariteit met het centrum maximaal is. Wanneer een mier die een object o_i aan het dragen is, zich op een vakje bevindt waarop de hoop T_j ligt, zal dit object op de hoop achtergelaten worden met een probabiliteit

$$P_d(o_i, T_j) = \begin{cases} 1 & \text{als } d(x_i, g_j) \leq d_g^*(T_j) \\ 1 - 0.9\min((\frac{d(x_i,g_j)}{d(O)^*})^{k_2}, 1) & \text{anders} \end{cases}$$

Hierbij is k_2 een (strikt) positief, natuurlijk getal. Wanneer de mier verschillende objecten aan het dragen is (de capaciteit van de mier is in dit geval groter dan 1), kunnen we de probabiliteit dat de hoop T_i wordt neergelegd op de hoop T_j analoog berekenen door in bovenstaande formule x_i te vervangen door het centrum van T_i.

Wanneer op een gegeven moment het aantal mieren groter wordt dan het aantal hopen, kan het gebeuren dat er geen enkele hoop nog aanwezig is op het rooster (wanneer de mieren slechts 1 object kunnen dragen, kan deze situatie niet voorkomen als er meer objecten dan mieren zijn). Om hieraan tegemoet te komen wordt aan elke mier a een zekere mate van geduld $p(a)$ geassocieerd. Wanneer de mier a reeds $p(a)$ verplaatsingen gemaakt heeft zonder de hoop die hij aan het dragen is te kunnen neerleggen, wordt de hoop neergelegd op het huidige vakje. We veronderstellen bovendien dat de mieren beschikken over een kortetermijngeheugen met dezelfde werking als bij het algoritme van Lumer en Faieta.

Om kleine classificatiefouten te verhelpen en om objecten die nog in verwerking zijn door een mier of zich nog geïsoleerd op het bord bevinden aan een cluster toe te wijzen, wordt na het mieralgoritme het k-gemiddelde (k-means) algoritme toegepast. De centra van de hopen

die door de mieren gevormd werden, worden gebruikt als initialisatie hiervoor. Men kan de geïsoleerde objecten ofwel toewijzen aan een cluster, ofwel geïsoleerd laten. Aangezien een object dat nog in verwerking is of geïsoleerd op het bord ligt op het einde van het algoritme een grote kans heeft om tot geen enkele cluster echt te behoren, wordt geopteerd om deze objecten ook voor het k-gemiddelde algoritme geïsoleerd te laten.

Wat betreft de capaciteit van de mieren zouden we kunnen gebruik maken van een heterogene populatie waarbij mieren met een verschillende capaciteit voorkomen. In [59] wordt er voor een andere strategie geopteerd: het algoritme start met een eerste waarde voor de parameters en wordt onderbroken en gevolgd door toepassing v.h. k-gemiddelde algoritme. Vervolgens wordt het mieralgoritme opnieuw toegepast, maar met een andere waarde voor de parameters, opnieuw gevolgd door k-gemiddelde. We kunnen dit zo een aantal keer na elkaar blijven toepassen voor steeds nieuwe parameters. Meestal wordt het algoritme een eerste keer uitgevoerd met mieren met capaciteit 1 en een tweede keer met mieren met oneindige capaciteit. De motivatie hiervoor is dat na de eerste uitvoering een groot aantal hechte clusters ontstaan. In de tweede uitvoering worden deze hechte clusters dan samengesmolten tot grotere clusters. In tegenstelling tot bij het algoritme van Lumer en Faieta krijgen we nu clusters waarbij de interclusterafstanden voldoende groot zijn.

In [60] wordt voorgesteld om voor de tweede uitvoering van het mieralgoritme de clusters uit de eerste uitvoering als atomaire objecten te beschouwen. Bovendien worden andere probabiliteiten beschouwd voor het opnemen in het geval er meerdere objecten op het vakje liggen:

- Als er een hoop ligt van 2 objecten, wordt willekeurig 1 van de 2 opgenomen met (vaste) probabiliteit $P_{destroy}$
- Als er een hoop T_j ligt met meer dan 2 objecten en centrum g_j wordt het meest verschillende object x weggenomen a.s.a.

$$\frac{d(x, g_j)}{\overline{d_g}(T_j)} > k_1 \tag{2.4}$$

Hierbij is k_1 een positieve, reële drempelwaarde. Ook voor het neerleggen worden andere probabiliteiten gebruikt:

- Als de cel leeg is wordt het object achtergelaten met een vaste probabiliteit P_{drop}
- Als de cel reeds een object x' bevat, wordt het object x neergelegd op x' a.s.a.

$$\frac{d(x, x')}{d^*(O)} < k_2 \tag{2.5}$$

- Als de cel reeds een hoop T_j met centrum g_j bevat, wordt het object x neergelegd op T_j a.s.a.

$$d(x, g_j) < d^*_g(T_j) \tag{2.6}$$

Hierbij is k_2 een positieve, reële drempelwaarde. In de tweede fase worden de centra van de hopen beschouwd bij de berekening van voorgaande probabiliteiten. Ook kunnen andere drempelwaarden gebruikt worden. Wanneer twee hopen op elkaar gelegd worden, wordt het resultaat, om de convergentiesnelheid te verhogen, opnieuw atomair beschouwd. In [44]

wordt voorgesteld het vage C-gemiddelde (fuzzy C-means) algoritme te gebruiken i.p.v. k-gemiddelde. Bovendien wordt vastgesteld dat het resulterende aantal clusters sterk afhankelijk is van de parameters. In het bijzonder blijkt de drempelwaarde die gebruikt wordt voor het samenvoegen van twee hopen in de tweede uitvoering van het algoritme, een sterke invloed uit te oefenen. Om hieraan enigszins tegemoet te komen wordt in [60] voorgesteld een populatie van heterogene mieren te gebruiken waarbij de mieren parameterwaarden hebben die uniform verdeeld zijn in een voorgedefinieerd interval. Tabel 2.1 vat de belangrijkste mijlpalen in de korte geschiedenis van miergebaseerde clusteringsalgoritmen nog eens samen.

jaar	auteur(s)	beschrijving
1990	Deneubourg et al.	agent-gebaseerd model voor aggregatie
1994	Lumer en Faieta	uitbreiding voor data-clusteren
1999	Monmarché	verschillende objecten per vakje

Tabel 2.1: Mijlpalen in de geschiedenis van miergebaseerde clusteringsalgoritmen

2.4 Alternatieve algoritmen

2.4.1 Herkenning van soortgenoten

In [53] wordt een clusteringsalgoritme voorgesteld dat geïnspireerd is op de manier waarop mieren hun nestgenoten herkennen. Iedere mier bezit een eigen lichaamsgeur die bepaald wordt door zowel het genoom van de mier als zijn omgeving. Telkens wanneer twee mieren elkaar tegenkomen vergelijken ze elkaars geur met het patroon dat ze in hun jeugd aangeleerd hebben. Dit patroon wordt gedurende hun hele leven aangepast en wordt de template van de mier genoemd. Voor elke (artificiële) mier i definiëren we nu $Label_i$ als het nummer van het nest waartoe zij behoort. Initieel behoren de mieren tot geen enkel nest ($Label_i = 0$ voor elke i). Het genoom van de mier correspondeert met een object uit de gegevensverzameling G. De template van de mier i is een drempelwaarde in het interval $[0, 1]$ en wordt voorgesteld door T_i. De eerste fase van het algoritme bestaat uit het initialiseren van de templates. In elke tijdstap worden 2 mieren uitgekozen die elkaar ontmoeten. Bij een dergelijke ontmoeting wordt de similariteit tussen de genomen van de twee mieren bepaald. Elke mier onthoudt de gemiddelde en de maximale similariteit met de mieren die hij reeds is tegengekomen. Noemen we $\overline{sim_i}$ de gemiddelde en sim_i^* de maximale similariteit die door mier i bepaald werd. In [53] wordt geargumenteerd, a.d.h.v. experimentele resultaten, dat het aantal ontmoetingen dat we moeten beschouwen in de eerste fase van het algoritme om goede resultaten te bekomen, lineair verwant is met het aantal mieren (m.a.w. met het aantal objecten). De template van deze mier wordt dan gegeven door [53]

$$T_i = \frac{\overline{sim_i} + sim_i^*}{2}$$

In de tweede fase van het algoritme worden in elke tijdstap opnieuw twee mieren uitgekozen die elkaar ontmoeten. We zeggen dat de twee mieren elkaar aanvaarden, genoteerd $A(i, j)$,

wanneer:

$$A(i,j) \Leftrightarrow (Sim(i,j) > T_i) \wedge (Sim(i,j) > T_j)$$

Hierbij is Sim een $G^2 - [0,1]$ afbeelding, waarbij voor objecten g_1 en g_2 uit G, $Sim(g_1, g_2)$ de mate van similariteit tussen g_1 en g_2 aangeeft. Elke mier houdt twee waarden bij, M_i en M_i^+. Deze waarden zijn initieel 0. Wanneer deze mier een mier ontmoet van hetzelfde nest, wordt de waarde M_i steeds geïncrementeerd en wordt de waarde M_i^+ geïncrementeerd wanneer de mieren elkaar aanvaarden. Wanneer deze mier een mier ontmoet van een ander nest worden de waarden M_i en M_i^+ beide met 1 verlaagd. De waarde M_i laat de mier toe om een schatting te maken van de grootte van zijn nest. Het toekennen of wijzigen van het nest van een mier gebeurt als volgt:

- Wanneer twee mieren, die beide nog tot geen enkel nest behoren, elkaar ontmoeten en aanvaarden, wordt een nieuw nest aangemaakt met initieel enkel deze twee mieren.
- Wanneer twee mieren elkaar ontmoeten en aanvaarden en slechts 1 van de mieren reeds tot een nest behoort, wordt de andere mier aan dit nest toegevoegd.
- Wanneer twee mieren uit hetzelfde nest elkaar ontmoeten, maar elkaar niet aanvaarden, wordt de minst geïntegreerde mier (de mier met de kleinste M^+ waarde) uit het nest verstoten.
- Wanneer twee mieren uit een ander nest elkaar ontmoeten en aanvaarden, wordt het kleinste nest (het nest van de mier met de kleinste M_i waarde) toegevoegd aan het grootste nest.

In [53] werd experimenteel vastgesteld dat het aantal iteraties dat beschouwd moet worden vooraleer convergentie optreedt lineair verwant is met het aantal objecten in de gegevensverzameling. Zowel de eerste als de tweede fase van het algoritme bestaan dus uit een lineair aantal stappen.

2.4.2 Levende structuren

In [2] wordt een algoritme beschreven dat gebaseerd is op het gedrag van de Argentijnse miersoort Linepithema humiles en de Afrikaanse soort Oerophylla longinoda. Bij deze soorten is recent vastgesteld dat mieren zich aan elkaar vasthechten om levende structuren te vormen die nuttig zijn in hun omgeving. Deze structuren variëren van boomstructuren tot kettingen van levende mieren om bijvoorbeeld takken van een boom te verbinden. Mieren kunnen over deze structuren wandelen en zich ergens vasthechten. Volgende eigenschappen worden benut door het algoritme:

- Er wordt op een vast punt gestart (bv. een tak, een blad ,...).
- De mieren kunnen zich op de gevormde structuur bewegen alsof het een vaste structuur is.
- De mieren kunnen zich eender waar vasthechten.
- Het grootste deel van de mieren is typisch geblokkeerd (bv. ze bevinden zich in het midden van een ketting).

- Een aantal mieren (bv. aan de uiteinden van een ketting) kan zich losmaken als ze willen.
- Men observeert zowel loskoppelingen als koppelingen.

Het algoritme stelt een boomstructuur op. De knopen worden gevormd door de objecten uit de gegevensverzameling G. Met elke mier correspondeert een knoop (en dus een object). We beschikken over een $G^2 - [0, 1]$ afbeelding die de similariteit tussen objecten van G weergeeft. Bovendien wordt met elke mier m_i $(i = 1, .., n)$ een similariteitsdrempelwaarde $S_{sim}(m_i)$ en een dissimilariteitsdrempelwaarde $S_{dissim}(m_i)$ geassocieerd. Initieel bevinden alle mieren zich op het vaste startpunt m_0. De drempelwaarden voor similariteit en dissimilariteit worden geinitialiseerd op 1 en 0 respectievelijk. Zolang er niet-vastgemaakte mieren bestaan, wordt het volgende herhaald:

- Als een mier m_i zich op het vast punt bevindt en de mier is voldoende (volgens zijn similariteitsdrempelwaarde) gelijkaardig aan de meest gelijkaardige mier m_+ die met het startpunt verbonden is, wandelt de mier m_i naar m_+ om in dezelfde klasse geklasseerd te worden. Anders, als m_i voldoende verschillend is (volgens zijn dissimilariteitsdrempelwaarde) van m_+, maakt m_i zich vast aan het startpunt. Als m_i onvoldoende gelijkaardig is en onvoldoende verschillend, worden de drempelwaarden aangepast zodat de mier toleranter is (de similariteitsdrempelwaarde wordt verminderd, de dissimilariteitswaarde verhoogd). De mier wordt dan opnieuw behandeld tijdens de volgende iteratie.
- Als een mier m_i zich op een andere mier m_{pos} bevindt, voldoende gelijkaardig is aan m_{pos} en voldoende verschillend is van de andere opvolgers van m_{pos}, dan hecht m_i zich aan m_{pos}. Op deze manier ontstaat een nieuwe subklasse van m_{pos} die maximaal verschillend is van de andere subklassen. In het andere geval worden de drempelwaarden aangepast en wandelt m_i naar een van de opvolgers van m_{pos}.

Merk op dat dit algoritme, in tegenstelling tot de andere algoritmen die we besproken hebben volledig deterministisch is. Bovendien is het resultaat afhankelijk van de volgorde waarin de objecten gepresenteerd worden.

2.4.3 Optimalisatie

Zonder twijfel de meest succesvolle toepassing van mieren houdt verband met combinatorische optimalisatie. Dorigo et al. introduceerden het gebruik van mieren voor het vinden van goede oplossingen voor het handelsreizigersprobleem [28, 29]. In het bijzonder inspireerden zij zich op de manier waarop mieren het kortste pad vinden naar een voedselbron aan de hand van spoorferomonen. Een oplossing van het handelsreizigersprobleem is een rondwandeling in een gewogen graaf van minimaal gewicht waarin elke knoop ten minste 1 keer is bevat. De knopen van de graaf worden steden genoemd en de gewichten van de bogen worden geïnterpreteerd als de afstanden tussen de steden. Het probleem wordt dan meestal geformuleerd in termen van een handelsreiziger die alle steden moet bezoeken door een minimale afstand af te leggen. Zoals wel bekend is dit probleem NP-Compleet, wat inhoudt dat een gegarandeerd optimale oplossing niet kan gevonden worden in aanvaardbare tijd. Om hieraan tegemoet te komen werden heuristieken voorgesteld die een sub-optimale oplossing vinden binnen een aanvaardbare tijd.

We schetsen hier kort het Ant System (AS) dat Dorigo et al. introduceerden voor het handelsreizigerprobleem. Gegeven zijn n steden, m mieren en de afstanden d_{ij} tussen stad i

en stad j $(i, j = 1, \ldots, n)$. De tijd is discreet en in elke tijdstap bevinden de mieren zich in een nieuwe stad. De mieren bepalen de volgende stad die ze zullen bezoeken op basis van de afstand naar deze stad, de aanwezige hoeveelheid feromonen op de boog die de huidige met de nieuwe stad verbindt, en een lijst met de steden die ze reeds bezocht hebben. Initieel wordt de feromonenintensiteit τ_{ij} op de boog die stad i met stad j verbindt, $(i, j \in \{1, 2, \ldots, n\}$, $i \neq j)$ gelijk gesteld aan een kleine constante. De probabiliteit dat de mier k die zich in stad i bevindt, zich begeeft naar stad j wordt gegeven door:

$$p_{ij} = \begin{cases} \dfrac{(\tau_{ij}(t))^{\alpha} \cdot (\eta_{ij})^{\beta}}{\sum_{l \notin S_k(i)} (\tau_{il}(t))^{\alpha} \cdot (\eta_{il})^{\beta}} & \text{als } j \notin S_k(i) \\ 0 & \text{anders} \end{cases}$$

Hierbij is $S_k(i)$ de lijst van reeds bezochte steden van mier k op het moment dat deze zich op knoop i bevindt, η_{ij} is de omgekeerde van de afstand tussen knopen i en j, en α en β zijn parameters die toelaten om het relatieve belang van de feromonen en de afstand in te stellen. Het aanpassen van de feromonen gebeurt wanneer alle mieren hun toer vervolledigd hebben:

$$\tau_{ij} \leftarrow \tau_{ij}(1 - \rho) + \sum_{k=1}^{m} \Delta\tau_{ij}^{k}$$

Met $\rho \in [0, 1]$ een parameter die de verdampingssnelheid van de feromonen weergeeft, $\Delta\tau_{ij}^{k}$ wordt gegeven door:

$$\Delta\tau_{ij}^{k} = \begin{cases} \frac{Q}{L^k} & \text{als de } k^{de} \text{ mier langs de boog } (i, j) \text{ gepasseerd is} \\ 0 & \text{anders} \end{cases}$$

Hierbij is Q een constante en L^k de lengte van de weg die mier k heeft afgelegd. Verschillende varianten voor het aanpassen van de feromonenintensiteit en de transitieprobabiliteiten werden bedacht. De meest succesvolle aanpak lijkt het Ant Colony System (ACS) te zijn [27]. De term Ant Colony Optimization (ACO) werd geïntroduceerd in [26] als verzamelnaam voor deze algoritmen. Hoewel deze ACO algoritmen reeds succesvol op een ruime klasse van problemen werden toegepast, gaande van routing in telecommunicatienetwerken [25] tot het opstellen van classificatieregels [64], ligt hun toepassing voor dataclustering niet voor de hand. In [78] wordt een variant van het Ant System gegeven voor het clusteren van data. Met elk object wordt een knoop in een graaf geassocieerd. Zoals bij het handelsreizigersprobleem wordt een pad in de graaf gezocht van minimaal gewicht. Het belangrijkste verschil is dat de mieren niet langer alle knopen moeten bezoeken. Initieel bezoeken de mieren ongeveer een tiende van het totale aantal knopen. Het aantal knopen dat de mieren moeten bezoeken wordt bovendien bij elke iteratie verminderd. De feromonenintensiteit op de bogen bepaalt ten slotte welke knopen, en dus welke objecten, tot dezelfde cluster behoren.

2.5 Conclusies

Miergebaseerde clusteringsalgoritmen bieden enkele belangrijke potentiële voordelen:

Parallellisering Aangezien enkel lokale evaluatie noodzakelijk is en er geen directe communicatie tussen de agenten geschiedt, lijkt parallellisatie van het algoritme niet uitgesloten. In welke mate de snelheidswinst die hiermee geassocieerd is, voldoende is om de

communicatie-overhead te compenseren, is echter onduidelijk. Bij een parallelle implementatie zal er immers voor gezorgd moeten worden dat geen 2 mieren tegelijk dezelfde hoop wijzigen.

Belangrijk hierbij is op te merken dat het clusteren dikwijls ten onrechte wordt toegeschreven aan zwermintelligentie (bijvoorbeeld in [8]). Zowel bij echte mieren als bij de meeste algoritmen die we besproken hebben, staat de grootte van de groep mieren in lineair verband met de snelheid waarmee de taak wordt uitgevoerd: n mieren zullen de taak n keer sneller uitvoeren dan 1 mier. Er is dus geen sprake van een samengesteld effect en dus evenmin van zwermintelligentie. Hoewel we duidelijk te maken hebben met een vorm van zelforganisatie, zijn er dus geen voordelen verbonden aan het gebruik van verschillende mieren bij implementaties op een monoprocessorsysteem.

Bij het algoritme van Lumer en Faieta kunnen problemen opduiken wanneer het aantal mieren te groot is. Er zijn dan onvoldoende objecten op het rooster aanwezig om de goede werking van het algoritme te kunnen garanderen. We kunnen dit gemakkelijk oplossen door het aantal mieren voldoende klein te houden. Wanneer mieren echter verschillende objecten tegelijk kunnen dragen zoals bij het algoritme van Monmarché [59] kunnen we dergelijke problemen niet vermijden.

Geen meta-informatie In tegenstelling tot nagenoeg alle klassieke clusteringsalgoritmen is geen informatie over de structuur van de gegevensverzameling, zoals bijvoorbeeld een initiële partitie of het aantal clusters, nodig. Nieuwe parameters werden echter geïntroduceerd. De schaalfactor α in (2.3) uit het algoritme van Lumer en Faieta heeft een grote invloed op het resultaat. Helaas blijkt deze parameter niet onafhankelijk te zijn van de aangeboden gegevensverzameling. In [37] wordt hier deels aan tegemoet gekomen door een vorm van leren in te voeren en de parameter aan te passen tijdens de uitvoering van het algoritme. Ook de waarde van drempelwaarden uit de ongelijkheden (2.4) en (2.5) zijn niet onafhankelijk van de gebruikte gegevensverzameling [44]. Kleine variaties in de waarden van deze parameters leiden tot grote verschillen in het resultaat.

Natuurlijke formulering Een formulering die gebruik maakt van mieren is natuurlijker en gemakkelijker te begrijpen dan bijvoorbeeld genetische algoritmen voor het clusteren van data. Enkel het vastleggen van eenvoudige lokale regels is vereist. Meestal wordt verondersteld dat ook echte mieren gebruik maken van vuistregels om te beslissen tot welke actie zij zullen overgaan [41]. De probabilistische aanpak zorgt voor de willekeur die fluctuaties mogelijk maakt waardoor de mieren zich kunnen aanpassen aan een onbekende en eventueel wijzigende omgeving. Ook dit is in overeenstemming met het geobserveerde gedrag van echte mieren.

Efficiëntie Het feit dat geen globale evaluatie van de data noodzakelijk is, suggereert dat deze algoritmen heel efficiënt kunnen gemaakt worden. Uitvoeringstijden die lineair zijn m.b.t. de grootte van de gegevensverzameling werden experimenteel vastgesteld in [53].

Niettemin kennen miergebaseerde clusteringsalgoritmen bijlange niet het succes van bijvoorbeeld ACO-algoritmen. Een belangrijk voordeel van de ACO-algoritmen is dat zij op een gepaste manier een abstractie maken van de principes die ervan aan de basis liggen. In essentie stelt een ACO-algoritme een probleem voor als een gewogen graaf en voorziet heuristieken om op basis van willekeurige oplossingen de gewichten aan te passen. De roostervoorstelling die nog steeds gebruikt wordt bij de meeste miergebaseerde clusteringsalgoritmen staat in fel

contrast hiermee. Deze voorstelling is afkomstig van het werk van Deneubourg die het gedrag van echte mieren wilde modelleren en bovendien een robotimplementatie op het oog had. Om te komen tot een efficiënt clusteringsalgoritme is een sterkere abstractie noodzakelijk. De beslissing van Monmarché om meerdere objecten op hetzelfde vakje toe te laten is duidelijk een stap in de goede richting aangezien de identificatie van afzonderlijke clusters nu mogelijk is en kan benut worden. De volgende stap is het volledig weglaten van het rooster. Hoewel dit reeds door Monmarché voorgesteld werd [59], werd dit door hem niet verder uitgewerkt. We zullen in hoofdstuk 4 verder bouwen op de ideeën van Monmarché om te komen tot een nieuw miergebaseerd clusteringsalgoritme.

Hoofdstuk 3

Wiskundige basisbegrippen

We bespreken in dit hoofdstuk enkele begrippen van wiskundige aard. Het miergebaseerd clusteringsalgoritme dat in het volgende hoofdstuk besproken zal worden, maakt gebruik van vaagregels. De combinatie van vaagverzamelingen, ruwverzamelingen en formele conceptanalyse, zal ons in staat stellen om documenten met elkaar te vergelijken. We beginnen met een overzicht van enkele bekende en minder bekende ordestructuren.

3.1 Ordestructuren

Definitie 1 (Preorde). *Een relatie R in een universum X die reflexief en transitief is, wordt een preorde genoemd.*

Definitie 2 (Equivalentierelatie). *Een symmetrische preorde R in een universum X wordt een equivalentierelatie genoemd.*

Definitie 3 (Tolerantierelatie). *Een relatie R in een universum X die reflexief en symmetrisch is, wordt een tolerantierelatie genoemd.*

Definitie 4 (Poset). *Een preorde $\leq$ in een universum X die antisymmetrisch is, wordt een partiële ordening genoemd. Het koppel $(X, \leq)$ noemt men een partieel geordende verzameling of kortweg poset.*

Definitie 5 (Ketting). *Een partieel geordende verzameling $(X, \leq)$ wordt een ketting genoemd als voldaan is aan*

$$(\forall (x, y) \in X^2)(x \leq y \text{ of } y \leq x)$$

Een ketting wordt soms ook nog een lineair of totaal geordende verzameling genoemd.

Definitie 6. *Zij $(X, \leq)$ een partieel geordende verzameling, $A \subseteq X$ en $b \in X$. Dan noemen we*

- *b een bovengrens voor A a.s.a. $(\forall a \in A)(a \leq b)$*
- *b een ondergrens voor A a.s.a. $(\forall a \in A)(b \leq a)$*
- *b het grootste element van A a.s.a. $b \in A$ en b is een bovengrens voor A*
- *b het kleinste element van A a.s.a. $b \in A$ en b is een ondergrens voor A*

- *b het supremum voor A a.s.a. b is de kleinste bovengrens voor A*
- *b het infimum voor A a.s.a. b is de grootste ondergrens voor A*

Definitie 7 (Tralie). *Een partieel geordende verzameling $(L, \leq)$ waarvoor elk doubleton een supremum en een infimum bezit, wordt een tralie (Eng. lattice) genoemd. Indien elke niet-ledige deelverzameling van L een supremum en een infimum bezit, noemen we de tralie compleet.*

Een tralie kan ook gedefinieerd worden als algebraïsche structuur i.p.v. als orde-structuur.

Definitie 8 (Tralie). *Een tralie is een drietal $(L, \vee, \wedge)$ met L een niet-ledige verzameling en $\vee$ en $\wedge$ binaire bewerkingen op L waarvoor voldaan is aan*

1. $(\forall a \in L)(a \vee a = a \text{ en } a \wedge a = a)$
2. $(\forall (a,b) \in L^2)(a \vee b = b \vee a \text{ en } a \wedge b = b \wedge a)$
3. $(\forall (a,b,c) \in L^3)((a \vee b) \vee c = a \vee (b \vee c) \text{ en } (a \wedge b) \wedge c = a \wedge (b \wedge c))$
4. $(\forall (a,b) \in L^2)(a \vee (a \wedge b) = a \text{ en } a \wedge (a \vee b) = a)$

Meestal wordt $\vee$ de join-bewerking en $\wedge$ de meet-bewerking genoemd. Voor het bewijs dat beide definities equivalent zijn verwijzen we naar [49]. De partiële ordening $\leq$ en de meet- en join-bewerking zijn verbonden m.b.v. volgende equivalentie

$$(\forall (a,b) \in L^2)(a \leq b \Leftrightarrow a \vee b = b \Leftrightarrow a \wedge b = a) \tag{3.1}$$

Bovendien geldt er voor a en b in L dat $a \wedge b = \inf\{a, b\}$ en $a \vee b = \sup\{a, b\}$.

Definitie 9. *[18] Zij $(L, \leq)$ een complete tralie. Als voor een willekeurige familie $(x_i)_{i \in I}$ in L voldaan is aan*

- $f(\sup\limits_{i \in I} x_i) = \sup\limits_{i \in I} f(x_i)$*, dan wordt f een supmorfisme genoemd*
- $f(\inf\limits_{i \in I} x_i) = \inf\limits_{i \in I} f(x_i)$*, dan wordt f een infmorfisme genoemd*
- $f(\sup\limits_{i \in I} x_i) = \inf\limits_{i \in I} f(x_i)$*, dan wordt f een duaal supmorfisme genoemd*
- $f(\inf\limits_{i \in I} x_i) = \sup\limits_{i \in I} f(x_i)$*, dan wordt f een duaal infmorfisme genoemd*

Eigenschap 1. *[20] Zij $(L, \leq)$ een complete tralie, f een stijgende $L - L$ afbeelding en g een dalende $L - L$ afbeelding, dan geldt voor elke familie $(x_i)_{i \in I}$*

$$\begin{aligned}
\sup_{i \in I} f(x_i) &\leq f(\sup_{i \in I} x_i) \\
\inf_{i \in I} f(x_i) &\geq f(\inf_{i \in I} x_i) \\
\sup_{i \in I} g(x_i) &\leq g(\inf_{i \in I} x_i) \\
\inf_{i \in I} g(x_i) &\geq g(\sup_{i \in I} x_i)
\end{aligned}$$

Definitie 10 (Adjunct paar). *[45] Zij $\leq_1$ een preorde in een universum X_1 en $\leq_2$ een preorde in een universum X_2, f een stijgende $X_2 - X_1$ afbeelding en g een stijgende $X_1 - X_2$ afbeelding. Het paar (f, g) wordt een adjunct paar genoemd als voldaan is aan*

$$(\forall (a, b) \in X_1 \times X_2)(f(b) \leq_1 a \Leftrightarrow b \leq_2 g(a))$$

Een adjunct paar wordt ook nog een Galois connectie, een veralgemeende inverse of een dialectische contradictie genoemd.

Definitie 11 (Monoïde). *Een monoïde is een drietal $(X, \otimes, e)$ bestaande uit een niet-ledige verzameling X, een associatieve bewerking $\otimes : X^2 \to X$ en een eenheidselement e in X voor deze bewerking.*

Definitie 12 (Partiële afbeelding). *[20] Zij X, Y en Z verzamelingen, f een $(X \times Y) - Z$ afbeelding en $(x_0, y_0) \in X \times Y$. De eerste partiële afbeelding van f in (x_0, y_0) is de $X - Z$ afbeelding $f(., y_0)$ gedefinieerd voor x in X als*

$$f(., y_0)(x) = f(x, y_0)$$

De tweede partiële afbeelding van f in (x_0, y_0) is de $Y - Z$ afbeelding $f(x_0, .)$ gedefinieerd voor y in Y als

$$f(x_0, .)(y) = f(x_0, y)$$

Definitie 13 (Monoïdale preorde). *[46] Een monoïdale preorde is een viertal $(X, \leq, \otimes, e)$ waarbij $(X, \otimes, e)$ een monoïde is, $\leq$ een preorde in X is en waarvoor de partiële afbeeldingen van $\otimes$ stijgend zijn.*

Definitie 14 (Gesloten preorde). *[46] Een gesloten preorde is een vijftal $(V, \leq, \otimes, \to, e)$ waarvoor $(V, \leq, \otimes, e)$ een monoïdale preorde is, $\to$ een $V^2 - V$ afbeelding is, $\otimes$ symmetrisch is en waarvoor $(\otimes(., v), \to (v, .))$ een adjunct paar vormt voor elke v in V. Hierbij is $\otimes(., v)$ de eerste partiële afbeelding van $\otimes$ in (v, v) en $\to (v, .)$ de tweede partiële afbeelding van $\to$ in (v, v).*

Meestal wordt V geïnterpreteerd als een verzameling van mogelijke waarheidswaarden. De gesloten preorde $(\{0, 1\}, \leq, \wedge, \Rightarrow, 1)$ komt bijvoorbeeld overeen met de klassieke binaire logica. De theorie van de vaagverzamelingen, die we verder zullen bespreken, maakt het mogelijk om wiskundige structuren, zoals verzamelingen en relaties, te veralgemenen door gradaties van lidmaatschap in een verzameling en gradaties van voldoen aan een relatie toe te laten. Gesloten preordes maken het mogelijk om dit principe te abstraheren.

Definitie 15 (Geresidueerde tralie). *[6] Een gesloten preorde $(V, \leq, \otimes, \to, e)$ waarvoor $(V, \leq)$ een complete tralie is, wordt een geresidueerde tralie genoemd.*

Het vijftal $(\mathbb{R}^+, \geq, +, -_b, 0)$ is een voorbeeld van een gesloten preorde die geen geresidueerde tralie is [46]. Hierbij is $\mathbb{R}^+$ de verzameling van alle niet-negatieve reële getallen en is $-_b$ de $\mathbb{R}^+ \times \mathbb{R}^+ - \mathbb{R}^+$ afbeelding gedefinieerd voor x en y in $\mathbb{R}^+$ als $\max(0, x - y)$.

Definitie 16 (V-ruimte). *[46] Zij $(V, \leq, \otimes, \to, e)$ een gesloten preorde, X een niet-ledige verzameling en μ een $X \times X - V$ afbeelding. Het koppel (X, μ) wordt een V-ruimte genoemd wanneer voldaan is aan*

reflexiviteit $(\forall x \in X)(e \leq \mu(x,x))$

transitiviteit $(\forall (x,y,z) \in X^3)(\mu(x,y) \otimes \mu(y,z) \leq \mu(x,z))$

de afbeelding μ noemt men de metriek op X en wordt meestal geassocieerd met een notie van afstand of similariteit. In de gesloten preorde $(\mathbb{R}^+, \geq, +, -_b, 0)$ herkennen we in de transitiviteitsvoorwaarde uit de voorgaande definitie de driehoeksongelijkheid. Een V-ruimte wordt symmetrisch genoemd als voldaan is aan

$$(\forall (x,y) \in X^2)(\mu(x,y) = \mu(y,x))$$

We veronderstellen verder steeds dat $(V, \leq, \otimes, \rightarrow, e)$ een gesloten preorde is. Een belangrijk voorbeeld van een V-ruimte is het koppel $(V, \rightarrow)$ [46].

Definitie 17 (V-map). *[46] Zij (X,μ) en (Y,ν) twee V-ruimten. Een $X - Y$ afbeelding f wordt een V-map genoemd als voldaan is aan*

$$(\forall (x,y) \in X^2)(\mu(x,y) \leq \nu(f(x), f(y)))$$

Een V-map A van een V-ruimte (X,μ) in de V-ruimte $(V,\rightarrow)$ kan worden geïnterpreteerd als een veralgemeende verzameling, waarbij $A(x)$ voor x in X, het lidmaatschap van x voorstelt in deze veralgemeende verzameling. Ook een relatie kan op deze manier veralgmeend worden.

Definitie 18 (V-relatie). *[46] Een V-relatie van een V-ruimte (X,μ) naar een V-ruimte (Y,ν) is een $X \times Y \rightarrow V$ afbeelding τ waarvoor voldaan is aan*

1. $(\forall (x,x') \in X^2)(\forall y \in Y)(\mu(x,x') \otimes \tau(x,y) \leq \tau(x',y))$
2. $(\forall x \in X)(\forall (y,y') \in Y^2)(\tau(x,y) \otimes \nu(y,y') \leq \tau(x,y'))$

Wegens de adjunctie van het paar $(\otimes, \rightarrow)$ is een V-relatie dus niets anders dan een $X \times Y \rightarrow V$ afbeelding, waarvan de partiële afbeeldingen respectievelijk V-maps tussen de V-ruimten (X,μ) en $(V,\rightarrow)$ en V-maps tussen de V-ruimten (Y,ν) en $(V,\rightarrow)$ zijn. Voor elke y in Y hebben we immers dat

$$(\forall (x,x') \in X^2)(\mu(x,x') \otimes \tau(x,y) \leq \tau(x',y))$$

equivalent is met

$$(\forall (x,x') \in X^2)(\mu(x,x') \leq \tau(x,y) \rightarrow \tau(x',y))$$

waaruit volgt dat $\tau(.,y)$ een V-map is tussen (X,μ) en $(V,\rightarrow)$. Een analoge redenering geldt voor de tweede partiële afbeeldingen van τ.

Zij X, Y en Z verzamelingen, R een (klassieke) relatie van X naar Y en S een klassieke relatie van Y naar Z. De compositie van R en S, genoteerd $R \circ S$, wordt gedefinieerd voor (x,z) in $X \times Z$ als

$$(x,z) \in R \circ S \Leftrightarrow (\exists y \in Y)((x,y) \in R \text{ en } (y,z) \in S)$$

Het is nu mogelijk om dit begrip uit te breiden naar V-relaties.

Definitie 19. *[46] Zij (X, μ), (Y, ν) en (Z, γ) drie V-ruimten en $\sigma : X \times Y \to V$ en $\tau : Y \times Z \to V$ twee V-relaties. De compositie van σ en τ is de $X \times Z \to V$ afbeelding $\sigma \circ \tau$, gedefinieerd voor $(x, z) \in X \times Z$ als*

$$(\sigma \circ \tau)(x, z) = \sup_{y \in Y}(\sigma(x, y) \otimes \tau(y, z))$$

Noteren we met Y^X de verzameling van alle V-maps van (X, μ) naar (Y, ν). Dan kunnen we Y^X de structuur geven van een V-ruimte m.b.v. de metriek μ, gedefinieerd voor f en g in Y^X als[1] [46]

$$\mu(f, g) = \inf_{x \in X} \nu(f(x), g(x))$$

Als bijzonder geval hiervan kunnen we de V-ruimte $(V^X, \to)$ beschouwen, die correspondeert met V-maps van (X, μ) naar $(V, \to)$, waarbij $\to$ voor ϕ en ψ in V^X gegeven wordt door [46]

$$\phi \to \psi = \inf_{x \in X}(\phi(x) \to \psi(x)) \tag{3.2}$$

Een element van V^X kan worden geïnterpreteerd als een veralgemeend predikaat.

Definitie 20 (V-predikaat). *[46] Een V-predikaat is een V-map van een V-ruimte (X, μ) in de V-ruimte $(V, \to)$.*

3.2 Operatoren op het eenheidsinterval

3.2.1 Triangulaire normen

Definitie 21 (Triangulaire norm). *Een triangulaire norm (of kortweg t-norm) is een stijgende, commutatieve en associatieve $[0, 1]^2 - [0, 1]$ afbeelding T waarvoor voldaan is aan de randvoorwaarde $T(x, 1) = x$, voor alle x in $[0, 1]$*

Het volgt onmiddellijk uit de definitie dat voor een willekeurige t-norm T voldaan is aan $T(0, 0) = T(0, 1) = T(1, 0) = 0$ en $T(1, 1) = 1$. Een t-norm valt over $\{0, 1\}^2$ dus samen met de klassieke conjunctie uit de binaire logica. Tabel 3.1 geeft een overzicht van enkele bekende t-normen.

Benaming	Beeldpuntdefinitie
Minimum	$T_M(x, y) = min(x, y)$
Product	$T_P(x, y) = x \cdot y$
Łukasiewicz	$T_W(x, y) = max(0, x + y - 1)$

Tabel 3.1: Enkele bekende triangulaire normen

Men gaat gemakkelijk na dat $(\forall(x, y) \in [0, 1]^2)(T_W(x, y) \leq T_P(x, y) \leq T_M(x, y))$ en dat T_M de enige idempotente t-norm is.

[1] We maken dus gebruik van overloading voor de definitie van μ

Eigenschap 2. *Voor een t-norm T en x in $[0,1]$ geldt $T(x,0)=0$*

Deze eigenschap volgt onmiddellijk uit $T(1,0)=0$ en het stijgend zijn van de eerste partiële afbeelding van T.

Definitie 22 (Nuldelers). *Zij T een t-norm en $(x,y)\in[0,1]^2$, dan worden x en y nuldelers genoemd van T a.s.a. $x\neq 0$, $y\neq 0$ en $T(x,y)=0$.*

Het is duidelijk dat T_M en T_P, in tegenstelling tot T_W, geen nuldelers hebben.

3.2.2 Triangulaire conormen

Definitie 23 (Triangulaire conorm). *Een triangulaire conorm (of kortweg t-conorm) is een stijgende, commutatieve en associatieve $[0,1]^2-[0,1]$ afbeelding S waarvoor voldaan is aan de randvoorwaarde $S(x,0)=x$, voor alle x in $[0,1]$*

Het volgt onmiddellijk uit de definitie dat voor een willekeurige t-conorm S voldaan is aan $S(1,1)=S(0,1)=S(1,0)=1$ en $S(0,0)=0$. Een t-conorm valt over $\{0,1\}^2$ dus samen met de klassieke disjunctie uit de binaire logica. Tabel 3.2 geeft een overzicht van enkele bekende t-conormen. Men gaat gemakkelijk na dat $(\forall(x,y)\in[0,1]^2)(S_M(x,y)\leq S_P(x,y)\leq S_W(x,y))$

Benaming	Beeldpuntdefinitie
Maximum	$S_M(x,y)=max(x,y)$
Probabilistische som	$S_P(x,y)=x+y-x\cdot y$
Begrensde som	$S_W(x,y)=min(1,x+y)$

Tabel 3.2: Enkele bekende triangulaire conormen

en dat S_M de enige idempotente t-conorm is.

Eigenschap 3. *Voor een t-conorm S en x in $[0,1]$ geldt $S(x,1)=1$*

Deze eigenschap volgt onmiddellijk uit $S(0,1)=1$ en het stijgend zijn van de eerste partiële afbeelding van S.

3.2.3 Negatoren

Definitie 24 (Negator). *Een negator is een dalende $[0,1]-[0,1]$ afbeelding N waarvoor $N(0)=1$ en $N(1)=0$. Als bovendien voor alle x in $[0,1]$ geldt dat $N(N(x))=x$, wordt deze negator involutief genoemd.*

Een negator valt over $\{0,1\}$ dus samen met de klassieke negatie uit de binaire logica. De bekendste negator is de $[0,1]-[0,1]$ afbeelding N_s gedefinieerd voor $x\in[0,1]$ als $N_s(x)=1-x$. We zullen naar deze negator verwijzen als de standaardnegator. Merk op dat deze standaardnegator involutief is.

Definitie 25. *Een t-norm T en een t-conorm S worden duaal genoemd t.o.v. een negator N a.s.a.*

$$(\forall(x,y)\in[0,1]^2)(S(x,y)=N(T(N(x),N(y))))$$

3.2.4 Implicatoren

Definitie 26 (Implicator). *Een implicator is een* $[0,1]^2 - [0,1]$ *afbeelding* I *met dalende eerste en stijgende tweede partiële afbeeldingen waarvoor* $I(0,0) = I(0,1) = I(1,1) = 1$ *en* $I(1,0) = 0$

Een implicator valt over $\{0,1\}^2$ dus samen met de klassieke implicatie uit de binaire logica.

Eigenschap 4. *Voor een willekeurige implicator* I *geldt* $I(0,x) = I(x,1) = 1$*, voor alle* x *in* $[0,1]$

Deze eigenschap volgt onmiddellijk uit $I(0,0) = 1$, $I(1,1) = 1$, het stijgend zijn van de tweede partiële afbeeldingen en het dalend zijn van de eerste partiële afbeeldingen.

Definitie 27 (Randimplicator). *Een randimplicator is een implicator* I *die voldoet aan*

$$(\forall x \in [0,1])(I(1,x) = x)$$

Eigenschap 5. *Voor een randimplicator* I *geldt* $I(x,y) \geq y$*, voor alle* x *en* y *in* $[0,1]$

Deze eigenschap volgt onmiddellijk uit de definitie van een randimplicator en het dalend zijn van de eerste partiële afbeeldingen van I.

De volgende eigenschap levert een methode om implicatoren te construeren, gebaseerd op de equivalentie van $P \Rightarrow Q$ en $\neg P \vee Q$ in de binaire logica.

Eigenschap 6. *Zij* S *een t-conorm en* N *een negator, de afbeelding* $I_{S,N}$*, gedefinieerd voor elementen* x *en* y *in* $[0,1]$ *als* $I_{S,N}(x,y) = S(N(x),y)$ *is een randimplicator. We noemen deze implicator de S-implicator geïnduceerd door* S *en* N*.*

Deze eigenschap volgt onmiddellijk uit de definities van negator, t-conorm en implicator. Tabel 3.3 geeft een overzicht van enkele S-implicatoren, geïnduceerd door de standaardnegator en een t-conorm.

Benaming	t-conorm	Beeldpuntdefinitie
Kleene-Dienes	S_M	$I_b(x,y) = max(1-x,y)$
Reichenbach	S_P	$I_r(x,y) = 1 - x + x \cdot y$
Łukasiewicz	S_W	$I_W(x,y) = min(1,1-x+y)$

Tabel 3.3: Enkele S-implicatoren geïnduceerd door de standaardnegator en een t-conorm

Definitie 28 (Rechterresidu). *[20] Het rechterresidu van een t-norm* T *is de* $[0,1]^2 - [0,1]$ *afbeelding* I_T*, gedefinieerd voor* x *en* y *in* $[0,1]$ *als*

$$I_T(x,y) = sup\{\lambda | \lambda \in [0,1] \text{ en } T(x,\lambda) \leq y\}$$

Eigenschap 7. *[20] Het rechterresidu* I_T *van een t-norm* T *is een randimplicator die we de residuele implicator (of R-implicator) van* T *noemen.*

Benaming	t-norm	Beeldpuntdefinitie
Gödel	T_M	$I_M(x,y) = \begin{cases} 1 & \text{als } x \leq y \\ y & \text{anders} \end{cases}$
Goguen	T_P	$I_P(x,y) = \begin{cases} 1 & \text{als } x \leq y \\ y/x & \text{anders} \end{cases}$
Łukasiewicz	T_W	$I_W(x,y) = min(1, 1-x+y)$

Tabel 3.4: Enkele residuele implicatoren

Met elke t-norm correspondeert er dus een implicator. Tabel 3.4 geeft een overzicht van enkele residuele implicatoren.

Eigenschap 8. *[20] Als de partiële afbeeldingen van een t-norm T supmorfismen zijn, zijn de eerste partiële afbeeldingen van I_T duale supmorfismen en de tweede partiële afbeeldingen van I_T infmorfismen.*

Eigenschap 9. *Voor een willekeurige t-norm T en zijn corresponderende residuele implicator I_T is voor a,b en c in $[0,1]$ voldaan aan*

$$a \leq b \quad \Rightarrow \quad I_T(a,b) = 1 \tag{3.3}$$

$$I(T(a,b),c) \quad \leq \quad I(a,I(b,c)) \tag{3.4}$$

Bewijs. Voor het bewijs van (3.4) verwijzen we naar [20]. Voor λ, a en b in $[0,1]$ en $a \leq b$ geldt steeds

$$T(\lambda, a) \leq T(1,a) = a \leq b$$

Bijgevolg is

$$\begin{aligned} I_T(a,b) &= sup\{\lambda | \lambda \in [0,1] \text{ en } T(a,\lambda) \leq b\} \\ &= sup\{\lambda | \lambda \in [0,1]\} \\ &= 1 \end{aligned}$$

en dus (3.3) □

Eigenschap 10. *[20] Zij T een t-norm en I een $[0,1]^2 - [0,1]$ afbeelding, er geldt: T en I voldoen aan de residueringsvoorwaarde*

$$(\forall (x,y,z) \in [0,1]^3)(T(x,y) \leq z \Leftrightarrow x \leq I(y,z)) \tag{3.5}$$

a.s.a. de partiële afbeeldingen van T sup-morfismen zijn en I het rechterresidu is van T.

In het bijzonder is aan de residueringsvoorwaarde (3.5) voldaan voor de t-normen en residuele implicatoren uit tabel 3.4 [20]. Voor een t-norm T en een implicator I die aan de residueringsvoorwaarde (3.5) voldoen vormt $(T(.,v), I(v,.))$ dus een adjunct paar voor elke $v \in [0,1]$. Wegens de definitie van een t-norm is $([0,1], T, 1)$ steeds een monoïde, is T symmetrisch en zijn de partiële afbeeldingen van T stijgend. Bovendien is $([0,1], \leq)$ een complete tralie. Bijgevolg is $([0,1], \leq, T, I, 1)$ een geresidueerde tralie voor elke t-norm T en $[0,1]^2 - [0,1]$ afbeelding I waarvoor voldaan is aan (3.5).

Eigenschap 11. *[20] Als T een t-norm is en I een $[0,1]^2 - [0,1]$ afbeelding zodat T en I voldoen aan de residueringsvoorwaarde* (3.5) *dan zijn de eerste partiële afbeeldingen van I duale supmorfismen en de tweede partiële afbeeldingen van I infmorfismen.*

Eigenschap 12. *Voor een t-norm T en een $[0,1]^2 - [0,1]$ afbeelding I waarvoor voldaan is aan de residueringsvoorwaarde* (3.5) *is voor $(a,b,c) \in [0,1]^3$ voldaan aan*

$$\begin{aligned}
a \leq b &\Leftrightarrow I(a,b) = 1 && (3.6)\\
T(a, I(b,c)) &\leq I(I(a,b), c) && (3.7)\\
T(I(a,b), c) &\leq I(a, T(b,c)) && (3.8)\\
a &\leq I(I(a,b), b) && (3.9)\\
T(a, I(a,b)) &\leq b && (3.10)\\
I(T(a,b), c) &= I(a, I(b,c)) = I(b, I(a,c)) && (3.11)\\
T(I(a,b), I(b,c)) &\leq I(a,c) && (3.12)\\
T(a,b) &\leq I(a,b) && (3.13)
\end{aligned}$$

Bewijs.

- Voor (3.6) moeten we wegens (3.3) enkel nog aantonen dat voor willekeurige a en b in $[0,1]$ voldaan is aan $I(a,b) = 1 \Rightarrow a \leq b$. We hebben achtereenvolgens

$$\begin{aligned}
I(a,b) = 1 &\Rightarrow 1 \leq I(a,b)\\
&\Leftrightarrow T(1,a) \leq b\\
&\Leftrightarrow a \leq b
\end{aligned}$$

  waarbij achtereenvolgens gebruik gemaakt is van de reflexiviteit van $\leq$, de residueringsvoorwaarde (3.5) en van de definitie van een t-norm.

- Voor het bewijs van (3.7) en (3.8) verwijzen we naar [20]

- Voor (3.9) hebben we wegens (3.7)

$$I(I(a,b), b) \geq T(a, I(b,b))$$

  Uit de reflexiviteit van $\leq$ en (3.6) volgt $I(b,b)$=1 en dus

$$I(I(a,b), b) \geq T(a,1) = a$$

- (3.10) volgt onmiddellijk uit (3.9) m.b.v. de residueringsvoorwaarde (3.5)

- Wegens eigenschap 10 is I een residuele implicator. Om (3.11) aan te tonen is het wegens de definitie van een residuele implicator nodig en voldoende om aan te tonen dat voor elke λ, a, b en c in $[0,1]$ geldt dat $T(T(a,b),\lambda) \leq c \Leftrightarrow T(a,\lambda) \leq I(b,c)$ en $T(T(a,b),\lambda) \leq c \Leftrightarrow T(b,\lambda) \leq I(a,c)$. Wegens de residueringsvoorwaarde (3.5) is dit equivalent met aantonen dat $T(T(a,b),\lambda) \leq c \Leftrightarrow T(T(a,\lambda),b) \leq c$ en $T(T(a,b),\lambda) \leq c \Leftrightarrow T(T(b,\lambda),a) \leq c$. Hieraan is wegens de associativiteit en commutativiteit van T steeds voldaan.

- Voor (3.12) vinden we wegens (3.7) dat

$$T(I(a,b), I(b,c)) \leq I(I(I(a,b),b),c)$$

  waaruit wegens (3.9) en het feit dat de eerste partiële afbeeldingen van een willekeurige implicator dalend zijn het gestelde volgt.

- Voor (3.13) ten slotte hebben we voor a en b in $[0,1]$ achtereenvolgens wegens de residueringsvoorwaarde (3.5), (3.11) en (3.6):

$$\begin{aligned} T(a,b) \leq I(a,b) &\Leftrightarrow a \leq I(b, I(a,b)) \\ &\Leftrightarrow a \leq I(a, I(b,b)) \\ &\Leftrightarrow a \leq I(a,1) = 1 \end{aligned}$$

  waaraan steeds is voldaan.

□

3.2.5 Aggregatie-operatoren

Definitie 29 (Aggregatie-operator). *[15] Een n-aire ($n \in \mathbb{N}^*$) aggregatie-operator is een continue, stijgende $[0,1]^n - [0,1]$ afbeelding $\oplus$ die voldoet aan $\oplus(0,\ldots,0) = 0$ en $\oplus(1,\ldots,1) = 1$. Bovendien wordt soms geëist dat $\oplus$ invariant is onder een permutatie van de argumenten en dat $(\forall a \in [0,1])(\oplus(a,\ldots,a) = a)$.*

3.3 Vaagverzamelingen

Definitie 30 (Vaagverzameling). *Een vaagverzameling in een universum X is een $X - [0,1]$ afbeelding.*

De klasse van alle vaagverzamelingen in X noteren we als $\mathcal{F}(X)$. Voor A in $\mathcal{F}(X)$ en x in X wordt $A(x)$ geïnterpreteerd als de mate waarin x tot de vaagverzameling A behoort, m.a.w. de mate waarin x voldoet aan het concept dat door A wordt gekarakteriseerd. We noemen $A(x)$ de lidmaatschapsgraad van x in A. Een klassieke verzameling wordt soms, om verwarring te vermijden, een scherpe verzameling genoemd.

Definitie 31. *Voor een vaagverzameling A in een universum X definiëren we de drager $supp(A)$, de kern $ker(A)$, de hoogte $hgt(A)$ en de plint $plt(A)$ als*

$$\begin{aligned} supp(A) &= \{x | x \in X \text{ en } A(x) > 0\} \\ ker(A) &= \{x | x \in X \text{ en } A(x) = 1\} \\ hgt(A) &= \sup_{x \in X} A(x) \\ plt(A) &= \inf_{x \in X} A(x) \end{aligned}$$

Definitie 32 (Zwak α-niveau). *Voor $A \in \mathcal{F}(X)$ en α in $]0,1]$ definiëren we het (zwak) α-niveau A_α van A als*

$$A_\alpha = \{x | x \in X \text{ en } A(x) \geq \alpha\}$$

Definitie 33 (Sterk α-niveau). *Voor $A \in \mathcal{F}(X)$ en α in $[0,1[$ definiëren we het sterk α-niveau $A_{\overline{\alpha}}$ van A als*

$$A_{\overline{\alpha}} = \{x | x \in X \text{ en } A(x) > \alpha\}$$

Eigenschap 13. *Zij $A \in \mathcal{F}(X)$ en $B \in \mathcal{F}(X)$. Er geldt*

$$A = B \Leftrightarrow (\forall \alpha \in]0,1])(A_\alpha = B_\alpha) \Leftrightarrow (\forall \alpha \in [0,1[)(A_{\overline{\alpha}} = B_{\overline{\alpha}})$$

Bewijs. We moeten enkel aantonen dat geldt $(\forall \alpha \in]0,1])(A_\alpha = B_\alpha) \Rightarrow A = B$ en $(\forall \alpha \in [0,1[)(A_{\overline{\alpha}} = B_{\overline{\alpha}}) \Rightarrow A = B$. We tonen de eerste implicatie aan; het bewijs van de tweede implicatie is volledig analoog. Onderstellen we dus dat geldt

$$(\forall \alpha \in]0,1])(A_\alpha = B_\alpha)$$

Zij $x \in X$, wegens de symmetrische rol van A en B kunnen we $A(x) \leq B(x)$ onderstellen. Stel nu dat $0 \leq A(x) < B(x)$. Dan zou $x \notin A_{B(x)}$ en $x \in B_{B(x)}$, een tegenstrijdigheid gezien de onderstelling en $B(x) \in]0,1]$. Bijgevolg moet $A(x) = B(x)$ gelden. □

De operatoren op het eenheidsinterval die we besproken hebben, laten toe om de belangrijkste bewerkingen op scherpe verzamelingen te veralgemenen naar vaagverzamelingen.

Definitie 34. *Zij T een t-norm, S een t-conorm, N een negator en A en B vaagverzamelingen in een universum X dan definiëren we de T-doorsnede $\cap_T$, de S-unie $\cup_S$ en het N-complement co_N als de vaagverzamelingen in X met beeldpuntdefinitie voor x in X gegeven door:*

$$\begin{aligned}(A \cap_T B)(x) &= T(A(x), B(x)) \\ (A \cup_S B)(x) &= S(A(x), B(x)) \\ (co_N A)(x) &= N(A(x))\end{aligned}$$

$A \cap_{T_M} B$ noteert men meestal als $A \cap B$ en noemt men de Zadeh doorsnede. Analoog bedoelen we met de Zadeh unie $A \cup B$ de vaagverzameling $A \cup_{S_M} B$ en zullen we meestal coA schrijven voor $co_{N_S} A$.

Definitie 35. *We definiëren de (scherpe) relatie $\subseteq$ in $\mathcal{F}(X)$ voor A en B vaagverzamelingen in X als*

$$A \subseteq B \Leftrightarrow A \cap B = A \Leftrightarrow (\forall x \in X)(A(x) \leq B(x))$$

Definitie 36. *Zij $(A_i)_{i \in I}$ een willekeurige familie van vaagverzamelingen in een universum X. We definiëren voor x in X*

$$\begin{aligned}(\bigcup_{i \in I} A_i)(x) &= \sup_{i \in I} A_i(x) \\ (\bigcap_{i \in I} A_i)(x) &= \inf_{i \in I} A_i(x)\end{aligned}$$

Eigenschap 14. *Zij $(A_i)_{i \in I}$ een willekeurige familie van vaagverzamelingen in een universum X. Voor i_0 in I en B een vaagverzameling in X geldt*

$$\bigcap_{i \in I} A_i \subseteq A_{i_0} \tag{3.14}$$

$$A_{i_0} \subseteq \bigcup_{i \in I} A_i \tag{3.15}$$

$$(\forall i \in I)(B \subseteq A_i) \Rightarrow (B \subseteq \bigcap_{i \in I} A_i) \tag{3.16}$$

$$(\forall i \in I)(A_i \subseteq B) \Rightarrow (\bigcup_{i \in I} A_i) \subseteq B \tag{3.17}$$

Deze eigenschappen volgen onmiddellijk uit de definitie van infimum en supremum.

Definitie 37 (Cartesiaans product). *Zij T een t-norm, A een vaagverzameling in een universum X en B een vaagverzameling in een universum Y. Het T-cartesiaans product van A en B is de vaagverzameling $A \times_T B$ in $X \times Y$, gedefinieerd voor (x, y) in $X \times Y$ als*

$$(A \times_T B)(x, y) = T(A(x), B(y))$$

Definitie 38 (Vaagrelatie). *Een vaagrelatie van een universum X naar een universum Y is een vaagverzameling in $X \times Y$.*

Verschillende belangrijke bewerkingen op klassieke relaties kunnen nu gemakkelijk worden uitgebreid naar vaagrelaties.

Definitie 39 (Inverse). *De inverse van een vaagrelatie R van een universum X naar een universum Y is de vaagrelatie R^{-1} van Y naar X, gedefinieerd voor x in X en y in Y als*

$$R^{-1}(y, x) = R(x, y)$$

Definitie 40 (Voor- en naverzameling). *[20] Zij $R \in \mathcal{F}(X \times Y)$, $x_0 \in X$ en $y_0 \in Y$. De R-voorverzameling van y_0 is de vaagverzameling Ry_0 in X gedefinieerd voor x in X als*

$$Ry_0(x) = R(x, y_0)$$

De R-naverzameling van x_0 is de vaagverzameling x_0R in Y gedefinieerd voor y in Y als

$$x_0R(y) = R(x_0, y)$$

Definitie 41 (Sup-T-compositie). *Zij R een vaagrelatie van een universum X naar een universum Y, zij S een vaagrelatie van Y naar een universum Z en zij T een t-norm. De sup-T-compositie van R en S is dan de vaagrelatie $R \circ_T S$ van X naar Z, gedefinieerd voor (x, z) in $X \times Z$ als*

$$(R \circ_T S)(x, z) = \sup_{y \in Y} T(R(x, y), S(y, z))$$

Definitie 42 (Subproduct). *[18] Zij R een vaagrelatie van een universum X naar een universum Y, S een vaagrelatie van Y naar een universum Z en I een implicator. Het subproduct van R en S is de vaagrelatie $R \lhd_I S$ van X naar Z, gedefinieerd voor x in X en z in Z als*

$$(R \lhd_I S)(x, z) = \inf_{y \in Y} I(R(x, y), S(y, z))$$

Definitie 43 (Superproduct). *[18] Zij R een vaagrelatie van een universum X naar een universum Y, S een vaagrelatie van Y naar een universum Z en I een implicator. Het superproduct van R en S is de vaagrelatie $R \rhd_I S$ van X naar Z, gedefinieerd voor x in X en z in Z als*

$$(R \rhd_I S)(x, z) = \inf_{y \in Y} I(S(y, z), R(x, y))$$

Voor een scherpe relatie R van X naar Y en A een (scherpe) deelverzameling van X is het direct beeld van A onder R de deelverzameling $R \uparrow A$ van Y, gedefinieerd door [21]

$$R \uparrow A = \{y | y \in Y \text{ en } (\exists x \in X)(x \in A \text{ en } (x, y) \in R)\}$$

Het direct beeld bevat dus alle elementen van Y die in relatie staan met minstens één element van A. Deze notie kan nu uitgebreid worden naar vaagrelaties.

Definitie 44 (Direct beeld). *[18] Zij R een vaagrelatie van een universum X naar een universum Y, A een vaagverzameling in X en T een t-norm. Het direct beeld van A onder R is de vaagverzameling $R \uparrow_T A$ in Y, gedefinieerd voor y in Y als*

$$(R \uparrow_T A)(y) = \sup_{x \in X} T(A(x), R(x, y))$$

Wanneer er geen gevaar is voor verwarring, zullen we de verwijzing naar de gebruikte t-norm T in $\uparrow_T$ weglaten.

Het superdirect beeld van een (scherpe) deelverzameling A van X onder een scherpe relatie R van X naar Y is de deelverzameling $R \downarrow A$ van Y, gedefinieerd door [21]

$$R \downarrow A = \{y | y \in Y \text{ en } Ry \subseteq A\}$$

Het superdirect beeld bevat dus alle elementen van Y die enkel in relatie staan met elementen van A. Indien voor een zekere y in Y geldt dat $Ry = \emptyset$, hebben we $y \in R \downarrow A$ voor een willekeurige deelverzameling A van X. Aangezien dit meestal niet gewenst is, wordt soms $(R \downarrow A) \cap (R \uparrow A)$ gebruikt als definitie voor het superdirect beeld (bijvoorbeeld in [48]). De notie van het superdirect beeld kan worden uitgebreid naar vaagrelaties.

Definitie 45 (Superdirect beeld). *[18] Zij R een vaagrelatie van een universum X naar een universum Y, A een vaagverzameling in X en I een implicator. Het superdirect beeld van A onder R is de vaagverzameling $R \downarrow_I A$ in Y, gedefinieerd voor y in Y als*

$$(R \downarrow_I A)(y) = \inf_{x \in X} I(R(x, y), A(x))$$

Dezelfde opmerking als voor het scherpe geval kan hier gemaakt worden. Wanneer geldt dat $(\forall y \in Y)(hgt(Ry) = 1)$ doen er zich geen problemen voor. In het andere geval zullen we ons er steeds moeten van vergewissen dat er geen ongewenste neveneffecten voorkomen bij waarden y uit Y waarvoor $hgt(Ry) < 1$. Wanneer er geen gevaar is voor verwarring, zullen we de verwijzing naar de gebruikte implicator I in $\downarrow_I$ weglaten.

Het subdirect beeld van een (scherpe) deelverzameling A van X onder een scherpe relatie R van X naar Y is de deelverzameling $R^{\triangleleft}(A)$ van Y, gedefinieerd door

$$R^{\triangleleft}(A) = \{y | y \in Y \text{ en } A \subseteq Ry\}$$

Het subdirect beeld bevat dus de elementen van Y die in relatie staan met alle elementen van A. Ook de notie van subdirect beeld kan worden uitgebreid naar vaagrelaties.

Definitie 46 (Subdirect beeld). *[18] Zij R een vaagrelatie van een universum X naar een universum Y, A een vaagverzameling in X en I een implicator. Het subdirect beeld van A onder R is de vaagverzameling $R^{\triangleleft_I}(A)$ in Y, gedefinieerd voor y in Y als*

$$R^{\triangleleft_I}(A)(y) = \inf_{x \in X} I(A(x), R(x, y))$$

Wanneer er geen gevaar voor verwarring is, zullen we de verwijzing naar de gebruikte implicator I in $^{\triangleleft_I}$ weglaten.

Ten slotte breiden we het begrip equivalentierelatie uit naar vaagrelaties m.b.v. een triangulaire norm.

Definitie 47. *Zij T een willekeurige t-norm. Een T-equivalentierelatie in een universum X is een vaagrelatie $E \in \mathcal{F}(X^2)$ waarvoor voldaan is aan*

reflexiviteit $(\forall x \in X)(E(x,x) = 1)$

symmetrie $(\forall x, y \in X)(E(x,y) = E(y,x))$

T-transitiviteit $(\forall x, y, z \in X)(T(E(x,y), E(y,z)) \leq E(x,z))$

Wanneer bovendien voldaan is aan $E(x,y) = 1 \Rightarrow x = y$ wordt E een T-gelijkheid genoemd.

3.4 Vaagregels

3.4.1 Vaagrestricties

Zij X een veranderlijke die waarden uit een universum U aanneemt en zij A een vaagverzameling in U. Een uitspraak van de vorm "X is A" wordt een vaagrestrictie op X genoemd. Voor u in U is $A(u)$ de mate waarin aan de opgelegde beperking wordt voldaan. Een possibiliteitsdistributie π_X voor een veranderlijke X is een $U - [0,1]$ afbeelding waarbij $\pi_X(u)$ voor u in U de mate voorstelt waarin u een mogelijke waarde is voor X. Wanneer enkel "X is A" gekend is, kunnen we $\pi_X = A$ stellen. Deze gelijkstelling staat in de literatuur bekend als de possibiliteitstoekenningsvergelijking [15].

Een andere mogelijkheid bestaat erin de uitspraak "X is A" te interpreteren als "X is A is zeker" of als " X is A is mogelijk". Het eerste geval kan in termen van possibiliteitsdistributies worden geïnterpreteerd als $(\forall u \in U)(\pi_X(u) \leq A(u))$. Met het tweede geval kunnen we dan de voorwaarde $(\forall u \in U)(\pi_X(u) \geq A(u))$ associëren [15]. Recent is men voor dit tweede geval gebruik beginnen maken van zogenaamde gegarandeerde possibiliteitsdistributies. Een dergelijke gegarandeerde possibiliteitsdistributie δ_X voor een veranderlijke X is een $U - [0,1]$ afbeelding waarbij $\delta_X(u)$ voor u in U de mate weergeeft waarin het gegarandeerd mogelijk is dat u een mogelijke waarde is voor X [16]. De uitspraak "X is A is mogelijk" wordt dan geïnterpreteerd als $(\forall u \in U)(\delta_X(u) \geq A(u))$.

Voor $\alpha \in [0,1]$ kunnen we dit uitbreiden naar vaagrestricties van de vorm " X is A is (ten minste) α-zeker". Hiermee correspondeert de voorwaarde [15]

$$(\forall u \in U)(\pi_X(u) \leq I_S(\alpha, A(u))) \tag{3.18}$$

waarbij I_S een willekeurige S-implicator is [15]. Met een vaagrestrictie van de vorm " X is A is (ten minste) α-mogelijk" correspondeert dan de voorwaarde [15][2]

$$(\forall u \in U)(\delta_X(u) \geq T(\alpha, A(u))) \tag{3.19}$$

waarbij T een willekeurige t-norm is.

3.4.2 Modelleren van vaagregels

Het verband tussen een veranderlijke X, die waarden aanneemt uit een universum U, en een veranderlijke Y die waarden aanneemt uit een universum V, kan vaak worden uitgedrukt m.b.v. regels van de vorm

$$\text{Als } X \text{ is } A \text{ dan } Y \text{ is } B \tag{3.20}$$

[2]Hoewel de uit [15] aangehaalde resultaten in [15] niet geformuleerd werden in termen van gegarandeerde possibiliteitsdistributies, blijven deze onverminderd geldig in deze context.

waarbij $A \in \mathcal{F}(U)$ en $B \in \mathcal{F}(V)$. Voor A en B scherpe verzamelingen herleidt dit zich tot een uitspraak van de vorm "Als $X \in A$ dan $Y \in B$". Noemen we R in $U \times V$ de verzameling van alle koppels van waarden voor (X, Y) die aan deze uitspraak voldoen. Het is duidelijk dat $A \times B \subseteq R \subseteq co(A \times coB)$. Beschouwen we nu A in $\mathcal{F}(U)$, B in $\mathcal{F}(V)$ en de regel "Als X is A dan Y is B". Dergelijke vaagregels zullen centraal staan bij de beschrijving van het miergebaseerd clusteringsalgoritme dat we in het volgende hoofdstuk zullen bespreken. Naar analogie met het scherpe geval, stellen we voor de gezamenlijke possibiliteitsdistributie $\pi_{X,Y}$ van X en Y, de gezamenlijke gegarandeerde possibiliteitsdistributie $\delta_{X,Y}$ van X en Y, $u \in U$ en $v \in V$

$$T(A(u), B(v)) \leq \delta_{X,Y}(u, v) \tag{3.21}$$

$$\pi_{X,Y}(u, v) \leq I(A(u), B(v)) \tag{3.22}$$

Wanneer we (3.21) opleggen spreekt men van het conjunctiegebaseerd model. Dit correspondeert met de interpretatie "Als X is A dan is het mogelijk dat Y is B". Wanneer we (3.22) opleggen, spreekt men over het implicatiegebaseerd model, wat correspondeert met de interpretatie "Als X is A dan is het zeker dat Y is B"

Voor het implicatiegebaseerd model onderscheiden we, naargelang de keuze van de gebruikte implicator, de volgende gevallen [30]:

Zekerheidsregels We kunnen (3.20) interpreteren als "Hoe meer X tot A behoort, hoe zekerder men kan zijn dat Y tot B behoort". In termen van vaagrestricties kunnen we dit uitdrukken als "Y is B is $A(u)$-zeker". Voor u in U hebben we dus de voorwaarde $(\forall v \in V)(\pi_Y(v) \leq I_S(A(u), B(v)))$ met I_S een S-implicator. Hieraan is steeds voldaan wanneer $(\forall (u, v) \in U \times V)(\pi_{X,Y}(u, v) \leq I_S(A(u), B(v)))$ [15].

Graduele regels De interpretatie "Hoe meer X tot A behoort, hoe meer Y tot B behoort" kunnen we beschrijven door voor u in U te eisen dat $(\forall v \in V)(T(A(u), \pi_{X,Y}(u, v)) \leq B(v))$ met T een t-norm waarvan de partiële afbeeldingen supmorfismen zijn. Wegens het verband tussen een dergelijke t-norm en zijn corresponderende R-implicator is hieraan steeds voldaan wanneer we eisen dat $(\forall (u, v) \in U \times V)(\pi_{X,Y}(u, v) \leq I_T(A(u), B(v)))$ waarbij I_T de R-implicator is die wordt voortgebracht door T.

Ook voor het conjunctiegebaseerd model kunnen we twee gevallen onderscheiden [30]:

Mogelijkheidsregels Beschouwen we de interpretatie "Hoe meer X tot A behoort, hoe meer het mogelijk wordt dat Y tot B behoort" wat overeenkomt met stellen dat 'Y is B is $A(u)$-mogelijk". Voor $u \in U$ hebben we dus de voorwaarde $(\forall v \in V)(T(A(u), B(v)) \leq \delta_Y(v))$ met T een t-norm. Hieraan is steeds voldaan wanneer $(\forall (u, v) \in U \times V)(T(A(u), B(v)) \leq \delta_{X,Y}(u, v))$ [15].

Antigraduele regels De interpretatie "Hoe meer X tot A behoort, hoe meer Y tot B behoort" kunnen we ook beschrijven door voor $u \in U$ te eisen dat $(\forall v \in V)(T(A(u), 1 - \delta_{X,Y}(u, v)) \leq 1 - B(v)))$, waarbij T een t-norm is waarvan de partiële afbeeldingen supmorfismen zijn. Wegens het verband tussen een dergelijke t-norm en zijn corresponderende R-implicator is hieraan steeds voldaan wanneer we eisen dat $(\forall (u, v) \in U \times V)(\delta_{X,Y}(u, v) \geq 1 - I_T(A(u), 1 - B(v)))$ waarbij I_T de R-implicator is die wordt voortgebracht door T.

Soms kan het gebeuren dat in het antecedent of het consequent van een vaagregel meerdere variabelen voorkomen, vb:

$$\text{Als } X_1 \text{ is } A_1 \text{ en } X_2 \text{ is } A_2 \text{ en } \ldots \text{en } X_m \text{ is } A_m \text{ dan } Y_1 \text{ is } B_1 \text{ en } \ldots \text{en } Y_n \text{ is } B_n \tag{3.23}$$

waarbij X_i een veranderlijke is die waarden aanneemt uit een universum U_i ($i \in \{1, 2, \ldots, m\}$), Y_j een veranderlijke is die waarden aanneemt uit een universum V_j ($j \in \{1, 2, \ldots, n\}$) en $A_i \in \mathcal{F}(U_i)$ en $B_j \in \mathcal{F}(V_j)$ voor i in $\{1, 2, \ldots, m\}$ en j in $\{1, 2, \ldots, n\}$. Hierbij zullen we steeds onderstellen dat zowel de veranderlijken in het antecedent als de veranderlijken in het consequent onafhankelijk zijn van elkaar. In dit geval kunnen we (3.23) beschouwen als een verzameling van n vaagregels van de vorm ($j \in \{1, \ldots, n\}$) [15]

$$\text{Als } (X_1, X_2, \ldots, X_m) \text{ is } A_1 \times_T A_2 \times_T \cdots \times_T A_n \text{ dan } Y_j \text{ is } B_j \tag{3.24}$$

waarbij T een t-norm is. Het geval met meerdere veranderlijken herleidt zich dus naar het geval met 1 veranderlijke in antecedent en consequent. We beschouwen derhalve verder enkel vaagregels van de vorm (3.20).

3.4.3 Inferentie met vaagregels

Beschouwen we de uitdrukkingen "X is A" en "X en Y zijn R", waarbij X een veranderlijke is die waarden aanneemt uit een universum U, Y een veranderlijke is die waarden aanneemt uit een universum V, $A \in \mathcal{F}(U)$ en $R \in \mathcal{F}(U \times V)$. De compositieregel voor inferentie stelt dat we hieruit kunnen afleiden dat "Y is $A \uparrow_T R$" voor een zekere t-norm T.

Beschouwen we nu een verzameling van k vaagregels waarbij de i^{de} vaagregel wordt gegeven door:

$$\text{Als } X \text{ is } A_i \text{ dan } Y \text{ is } B_i \tag{3.25}$$

waarbij $A_i \in \mathcal{F}(U)$ en $B_i \in \mathcal{F}(V)$ voor i in $\{1, 2, \ldots, k\}$, en waarbij X en Y veranderlijken zijn die respectievelijk waarden aannemen uit het universum U en V. Onderstellen we nu dat de i^{de} vaagregel wordt voorgesteld d.m.v. een vaagrelatie R_i van U naar V. Wegens de compositieregel voor inferentie kunnen we uitgaande van een waarneming "X is A' " en de i^{de} vaagregel besluiten dat "Y is $A' \uparrow_T R_i$", waarbij T een t-norm is. Om nu informatie over de volledige verzameling vaagregels te betrekken bij het inferentieproces, maken we gebruik van een aggregatie-operator $\oplus$. We beschikken nu over twee mogelijkheden om kennis omtrent Y af te leiden op basis van de waarneming "X is A' " [15]. De eerste mogelijkheid die FITA (Eng. First Infer Then Aggregate) genoemd wordt, wordt gegeven door

$$Y \text{ is } \oplus_{i=1}^{k}(A' \uparrow_T R_i) \tag{3.26}$$

De tweede mogelijkheid wordt FATI (Eng. First Aggregate Then Infer) genoemd; we krijgen hiervoor

$$Y \text{ is } A' \uparrow_T \oplus_{i=1}^{k} R_i \tag{3.27}$$

Wanneer we in het scherpe geval de uitspraken "Als $X \in A_1$ dan $Y \in B_1$" en "Als $X \in A_2$ dan $Y \in B_2$" beschouwen, krijgen we in het conjunctiegebaseerd model dat voor $X \in A_1 \cap A_2$ alle waarden in $B_1 \cup B_2$ mogelijk zijn voor Y [30]. In het implicatiegebaseerd model krijgen we dat de waarde van Y zeker in $B_1 \cap B_2$ gelegen is [30]. Een voor de hand liggende uitbreiding naar vaagregels bestaat er dan in om bij het conjunctiegebaseerd model, een continue t-conorm te gebruiken als aggregatieoperator en bij het implicatiegebaseerd model een continue t-norm

te gebruiken. Een in de praktijk veel voorkomende keuze voor het conjunctiegebaseerd model bestaat erin de vaagregels te interpreteren m.b.v. het minimum en de regels te aggregeren m.b.v. het maximum. In dit geval valt FATI samen met FITA [15]. Dit zal ook de keuze zijn waar we in het volgende hoofdstuk zullen mee werken. Ten slotte dient nog worden opgemerkt dat er in het conjunctiegebaseerd model eigenlijk geen sprake is van inferentie, aangezien de vaagregel gemodelleerd wordt d.m.v. een t-norm. Het conjunctiegebaseerd model kan worden opgevat als een manier om een functie die niet exact gekend is, te beschrijven. Een vaagregel van de vorm (3.20) wordt dan opgevat als de beschrijving van een zogenaamd vaagpunt $A \times_T B$ [30]. Het beschreven inferentiemechanisme voor een verzameling vaagregels, is dan in feite een interpolatiemechanisme voor een verzameling van dergelijke vaagpunten.

3.4.4 Defuzzificatie

Vaagregels zijn bijgevolg geschikt om uitgaande van een waarneming omtrent een veranderlijke X, kennis af te leiden omtrent de mogelijke waarden van een andere veranderlijke Y. In praktische toepassingen is het meestal handiger om over 1 scherpe waarde te beschikken die de veranderlijke Y karakteriseert. Een defuzzificatie-operator komt hieraan tegemoet. We beperken ons hier tot het geven van een voor de praktijk belangrijk voorbeeld: de zwaartepuntmethode.

Definitie 48 (Zwaartepuntmethode). *[15] Zij A een vaagverzameling in een deelinterval $U = [u_1, u_2]$ van $\mathbb{R}$. De zwaartepuntmethode associeert met A de scherpe waarde $COG(A)$ gedefinieerd door*

$$\begin{aligned} COG: \quad & \mathcal{F}(U) \to U \\ & \emptyset \mapsto u_1 \\ & A \mapsto \frac{\sum_{u=u1}^{u_2} uA(u)}{\sum_{u=u1}^{u_2} A(u)}, \forall A \in \mathcal{F}(U) \setminus \{\emptyset\} \end{aligned}$$

3.5 Ruwverzamelingen

3.5.1 Inleiding

Veronderstel dat een bepaald concept in een universum U enkel gekarakteriseerd is door een verzameling instanties $A \subseteq U$. Zij $E \subseteq U \times U$ een relatie die het niet onderscheidbaar zijn van objecten uit het universum definieert, m.a.w. twee instanties u_1 en u_2 uit U zijn niet onderscheidbaar van elkaar a.s.a. $(u_1, u_2) \in E$. We noemen een dergelijke relatie een ononderscheidbaarheidsrelatie (Eng. indiscernibility relation). Normaal wordt geëist dat E een equivalentierelatie is.

Definitie 49 (Benaderingsruimte). *Zij U een universum en E een equivalentierelatie in U. Het koppel (U, E) wordt een benaderingsruimte genoemd (Eng. approximation space).*

Definitie 50 (Onder- en bovenbenadering). *Zij (U, E) een benaderingsruimte en A een deelverzameling van U. De onderbenadering $\underline{E}A$ en de bovenbenadering $\overline{E}A$ van A in (U, E)*

zijn respectievelijk gegeven door [20]

$$\begin{aligned} \underline{E}A &= E \downarrow A = \{x | x \in U \text{ en } Ex \subseteq A\} \\ \overline{E}A &= E \uparrow A = \{x | x \in U \text{ en } Ex \cap A \neq \emptyset\} \end{aligned}$$

Wanneer $x \in \underline{E}A$ zeggen we dat x noodzakelijk een instantie is van het concept; wanneer $x \in \overline{E}A$ zeggen we dat x mogelijk een instantie is van het concept. Het koppel $(\underline{E}A, \overline{E}A)$ wordt de ruwverzameling voor A genoemd [20]. Als E een equivalentierelatie is, deelt E het universum op in equivalentieklassen. Noem U/E de verzameling van alle equivalentieklassen geïnduceerd door E. De onderbenadering van E is dan de grootste deelverzameling $X \subseteq U/E$ van equivalentieklassen waarvoor $\bigcup X \subseteq A$. Analoog is de bovenbenadering van E dan de kleinste deelverzameling $X \subseteq U/E$ van equivalentieklassen waarvoor $A \subseteq \bigcup X$. Ruwverzamelingen werden in 1982 ingevoerd door Pawlak [65]. Het verband tussen onderbenadering resp. bovenbenadering en superdirect resp. direct beeld suggereert volgende uitbreiding naar vaagrelaties.

Definitie 51 (Vaagbenaderingsruimte). *Zij U een universum, T een t-norm en E een T-equivalentierelatie in U. Het koppel (U, E) wordt een vaagbenaderingsruimte genoemd.*

Definitie 52 (Onder- en bovenbenadering). *[70] Zij (U, E) een vaagbenaderingsruimte en A een vaagverzameling in U, zij I een implicator en T een t-norm. De onder- en bovenbenadering van A in (U, E) worden respectievelijk gegeven door*

$$\begin{aligned} \underline{E}A(y) &= (E \downarrow_I A)(y) = \inf_{x \in U} I(E(x, y), A(x)) \\ \overline{E}A(y) &= (E \uparrow_T A)(y) = \sup_{x \in U} T(A(x), E(x, y)) \end{aligned}$$

Het koppel $(\underline{E}A, \overline{E}A)$ wordt een ruwvaagverzameling (Eng. rough fuzzy set) voor A genoemd wanneer E scherp en A vaag zijn, en een vaagruwverzameling (Eng. fuzzy rough set) wanneer E vaag en A scherp zijn [76]. We zullen de term vaagruwverzameling verder eveneens gebruiken voor het geval waarbij zowel A als E vaag zijn. Een andere mogelijke uitbreiding van ruwverzamelingen, die gesuggereerd werd in [46], is door gebruik te maken van een geresidueerde tralie.

Definitie 53 (Onder- en bovenbenadering). *Zij $(V, \leq, \otimes, \rightarrow, e)$ een geresidueerde tralie, (U, μ) een symmetrische V-ruimte en A een V-map van (U, μ) naar $(V, \rightarrow)$. De onder- en bovenbenadering van A in (U, μ) worden respectievelijk gegeven door*

$$\begin{aligned} \underline{E}A(y) &= \inf_{x \in U} (\mu(x, y) \rightarrow A(x)) \\ \overline{E}A(y) &= \sup_{x \in U} (A(x) \otimes \mu(x, y)) \end{aligned}$$

3.5.2 Veralgemeende ruwverzamelingen en modale logica

Wanneer we afstappen van de eis dat de ononderscheidbaarheidsrelatie symmetrisch moet zijn, worden we geconfronteerd met twee alternatieven voor het berekenen van onder- en bovenbenaderingen. Zij R een (scherpe) niet-symmetrische ononderscheidbaarheidsrelatie in U. Enerzijds zouden we de bovenbenadering van een deelverzameling A kunnen berekenen als $R \uparrow A$ zoals in het symmetrische geval. Anderzijds is ook $R^{-1} \uparrow A$ een veralgemening van het

symmetrische geval. Een analoge opmerking kan gemaakt worden voor onderbenaderingen en de uitbreidingen naar vaagrelaties en V-relaties. Om te bepalen welke van deze twee alternatieven zinvol gedefinieerd kan worden en wat de semantiek is die we kunnen hechten aan niet-symmetrische ononderscheidbaarheidsrelaties, doen we een beroep op het verband dat bestaat tussen ruwverzamelingen en modale logica.

We schetsen kort het begrip modale logica en het verwantschap met ruwverzamelingen [83]. Zij Φ de niet-ledige verzameling die alle proposities bevat die kunnen geconstrueerd worden, uitgaande van de propositionele constanten T en F, een aftelbare verzameling propositionele veranderlijken $P = \{a, b, \dots\}$ en de logische connectieven $\vee, \wedge, \rightarrow$ en $\neg$. Zij W een niet-ledige verzameling waarvan de elementen werelden genoemd worden, $R \subseteq W \times W$ een binaire relatie die de bereikbaarheidsrelatie (Eng. accessibility relation) genoemd wordt. Voor w en w' in W, zeggen we dat w' bereikbaar is vanuit w wanneer $(w, w') \in R$. Een interpretatie in (W, R) is een afbeelding $v : W \times P \rightarrow \{\text{waar}, \text{vals}\}$. Voor een propositie a in P en een wereld w in W, zeggen we dat a waar is in w onder de interpretatie v als voldaan is aan $v(w, a) = \text{waar}$ en noteren dit als $w \vDash_v a$; $v(w, a) = \text{vals}$ noteren we als $w \nvDash_v a$. De uitbreiding van v tot een valuatie voor willekeurige proposities $v^* : W \times \Phi \rightarrow \{\text{waar}, \text{vals}\}$, gebeurt op de gebruikelijke manier. Voor a in P geldt $w \vDash_{v^*} a$ a.s.a $w \vDash_v a$, voor ϕ en ψ in Φ geldt

$$\begin{array}{rll}
w \vDash_{v^*} (\phi \wedge \psi) & \text{a.s.a.} & w \vDash_{v^*} (\phi) \text{ en } w \vDash_{v^*} (\psi) \\
w \vDash_{v^*} (\phi \vee \psi) & \text{a.s.a.} & w \vDash_{v^*} (\phi) \text{ of } w \vDash_{v^*} (\psi) \\
w \vDash_{v^*} (\neg\phi) & \text{a.s.a.} & w \nvDash_{v^*} (\phi) \\
w \vDash_{v^*} (\phi \rightarrow \psi) & \text{a.s.a.} & w \nvDash_{v^*} (\phi) \text{ of } w \vDash_{v^*} (\psi)
\end{array}$$

Naast deze standaard connectieven beschikt de modale logica nog over de noodzakelijkheidsoperator $\Box$ en de mogelijkheidsoperator $\Diamond$, gedefinieerd door:

$$\begin{array}{l}
w \vDash_{v^*} \Box\phi \text{ a.s.a. } (\forall w' \in W)((w, w') \in R \Rightarrow w' \vDash \phi) \\
w \vDash_{v^*} \Diamond\phi \text{ a.s.a. } (\exists w' \in W)((w, w') \in R \text{ en } w' \vDash \phi)
\end{array}$$

We noemen de propositie ϕ noodzakelijk waar in de wereld w onder de interpretatie v als $v(\Box\phi) = \text{waar}$, m.a.w. als ϕ waar is onder de interpretatie v in elke wereld die toegankelijk is vanuit w . We noemen ϕ mogelijk waar in de wereld w als onder de interpretatie v als $v(\Diamond\phi) = \text{waar}$, m.a.w. als ϕ waar is onder de interpretatie v in een wereld die bereikbaar is vanuit w . We kunnen nu met elke propositie de verzameling associëren van alle werelden waarin de beschouwde propositie waar is. Definiëren we hiertoe $t : \Phi \rightarrow \mathcal{P}(W)$ als volgt:

$$t(\phi) = \{w | w \in W \text{ en } w \vDash \phi\}$$

Hiervan gebruik makend kunnen we de klassieke logische connectieven interpreteren als operaties op verzamelingen. Zo hebben we bijvoorbeeld $t(\phi \wedge \psi) = t(\phi) \cap t(\psi)$ en $t(\neg\phi) = co\, t(\phi)$. Voor de operatoren $\Box$ en $\Diamond$ krijgen we:

$$\begin{array}{rcl}
t(\Box\phi) & = & \{x | x \in U \text{ en } xR \subseteq t(\phi)\} \\
t(\Diamond\phi) & = & \{x | x \in U \text{ en } xR \cap t(\phi) \neq \emptyset\}
\end{array}$$

Als R een equivalentierelatie is hebben we dus, wegens de symmetrie van R

$$\begin{aligned} t(\Box\phi) &= \underline{R}(t(\phi)) \\ t(\Diamond\phi) &= \overline{R}(t(\phi)) \end{aligned}$$

De modale logica wordt opgedeeld in categorieën op basis van de eigenschappen van de bereikbaarheidsrelatie R. Wanneer R een equivalentierelatie is, wordt de corresponderende logica S5 modale logica genoemd. De ruwverzamelingen zoals ze geïntroduceerd werden door Pawlak, corresponderen bijgevolg met deze S5 modale logica. Wanneer R geen equivalentierelatie is, correspondeert er met de bijhorende modale logica een veralgemeende benaderingsruimte. Niet-symmetrische ononderscheidbaarheidsrelaties kunnen we dus interpreteren als bereikbaarheidsrelaties.

Zij R bijvoorbeeld een relatie in U die het specifieker zijn van objecten modelleert, m.a.w. voor x en y in U interpreteren we $(x, y) \in R$ als "x is specifieker dan y". Een element x van A zal dan tot de onderbenadering van A behoren als ieder element dat bereikbaar is vanuit x, m.a.w. ieder object y waarvoor x specifieker is dan y, eveneens tot A behoort. Een element x van U zal tot de bovenbenadering van A behoren als er een element y van A bereikbaar is vanuit x, m.a.w. als x specifieker is dan een element uit A. Meer algemeen beschouwen we de volgende definities.

Definitie 54 (Veralgemeende benaderingsruimte). *Zij U een universum en R een willekeurige binaire relatie op U. Het koppel (U, R) wordt een veralgemeende benaderingsruimte genoemd.*

Definitie 55 (Onder- en bovenbenadering). *Zij (U, R) een veralgemeende benaderingsruimte en A een deelverzameling van U. De onderbenadering $\underline{R}A$ en een bovenbenadering $\overline{R}A$ van A in (U, R) zijn respectievelijk gegeven door:*

$$\begin{aligned} \underline{R}A &= R^{-1} \downarrow A = \{x | x \in U \text{ en } xR \subseteq A\} \\ \overline{R}A &= R^{-1} \uparrow A = \{x | x \in U \text{ en } xR \cap A \neq \emptyset\} \end{aligned}$$

Ook voor vaagruwverzamelingen kan een dergelijke veralgemening plaatsvinden. In [76] wordt een vaagmodale logica gedefinieerd, gebaseerd op uitbreidingen van de operatoren $\Box$ en $\Diamond$. Op dezelfde manier volgt dan de volgende keuze voor de definitie van onder- en bovenbenadering.

Definitie 56 (Veralgemeende vaagbenaderingsruimte). *Zij U een universum en R een binaire vaagrelatie in U. Het koppel (U, R) wordt een veralgemeende vaagbenaderingsruimte genoemd.*

Definitie 57 (Onder- en bovenbenadering). *Zij (U, R) een veralgemeende vaagbenaderingsruimte en A een vaagverzameling in U, zij I een implicator en T een t-norm. De onder- en bovenbenadering van A in (U, R) worden respectievelijk gegeven door*

$$\begin{aligned} \underline{R}A(y) &= (R^{-1} \downarrow_I A)(y) = \inf_{x \in U} I(R(y, x), A(x)) \\ \overline{R}A(y) &= (R^{-1} \uparrow_T A)(y) = \sup_{x \in U} T(A(x), R(y, x)) \end{aligned}$$

3.6 Formele Conceptanalyse

3.6.1 Inleiding

Objecten worden dikwijls beschreven d.m.v. een verzameling attributen of kenmerken. Zij G (Duits: Gegenstände) een verzameling objecten, M (Duits: Merkmale) een verzameling attributen en R een (scherpe) relatie van G naar M. Voor g in G en m in M, hechten we aan $(g, m) \in R$ de interpretatie dat g het attribuut m bezit, of nog, dat g het kenmerk m vertoont.

Definitie 58 (Context). *Een drietal (G, M, R), bestaande uit twee niet-ledige verzamelingen G en M en een relatie R van G naar M, wordt een (formele) context genoemd.*

Een concept wordt traditioneel gedefinieerd door extensie (de verzameling van alle objecten die een instantie zijn van het concept) of door intensie (de verzameling attributen waaraan het concept voldoet).

Definitie 59. *Zij (G, M, R) een formele context. De operatoren $^{\rightarrow_R} : \mathcal{P}(G) \rightarrow \mathcal{P}(M)$ en $^{\leftarrow_R} : \mathcal{P}(M) \rightarrow \mathcal{P}(G)$ worden gedefinieerd voor $A \subseteq G$ en $B \subseteq M$ als*

$$\begin{aligned} A^{\rightarrow_R} &= \{m | m \in M \text{ en } (\forall g \in A)((g, m) \in R)\} = \bigcap_{g \in A} gR = R^{\triangleleft}(A) \\ B^{\leftarrow_R} &= \{g | g \in G \text{ en } (\forall m \in B)((g, m) \in R)\} = \bigcap_{m \in B} Rm = (R^{-1})^{\triangleleft}(B) \end{aligned}$$

$A^{\rightarrow_R}$ is dus de verzameling van attributen die gedeeld worden door alle objecten in A, $B^{\leftarrow_R}$ is de verzameling van objecten die alle attributen uit B bezitten. Wanneer er geen gevaar is voor verwarring zullen we de verwijzing naar R in $^{\rightarrow_R}$ en $^{\leftarrow_R}$ weglaten.

Definitie 60 (Concept). *Een (formeel) concept in een formele context (G, M, R) is een koppel (A, B) met $A \subseteq G$, $B \subseteq M$, waarvoor voldaan is aan $A^{\rightarrow} = B$ en $B^{\leftarrow} = A$. De klasse van alle concepten in de context (G, M, R) noteren we als $\mathcal{B}(G, M, R)$ (Duits: Begriffsverband).*

Formele conceptanalyse werd in 1982 ingevoerd door Wille [82]. We vermelden tot slot nog zonder bewijs de centrale stelling van de formele conceptanalyse. Het bewijs zal volgen uit het bewijs van een analoge eigenschap voor vaagconcepten.

Stelling 1. *Zij (G, M, R) een formele context. De structuur $(\mathcal{B}(G, M, R), \vee, \wedge)$ is een complete tralie waarbij de join- en meet-bewerking respectievelijk gedefinieerd worden voor een familie $(A_i, B_i)_{i \in I}$ van concepten in (G, M, R) als*

$$\begin{aligned} \sup_{i \in I}(A_i, B_i) &= ((\bigcup_{i \in I} A_i)^{\rightarrow\leftarrow}, \bigcap_{i \in I} B_i) \\ \inf_{i \in I}(A_i, B_i) &= (\bigcap_{i \in I} A_i, (\bigcup_{i \in I} B_i)^{\leftarrow\rightarrow}) \end{aligned}$$

De met deze tralie corresponderende orderelatie wordt voor concepten (A, B) en (C, D) in $\mathcal{B}(G, M, R)$ gegeven door

$$(A, B) \leq (C, D) \Leftrightarrow A \subseteq C \Leftrightarrow B \supseteq D$$

3.6.2 Vaagconcepten

Definitie 61 (V-context). *Zij $(V, \leq, \otimes, \rightarrow, e)$ een geresidueerde tralie, (G, μ) en (M, ν) twee V-ruimten en R een V-relatie van (G, μ) naar (M, ν). Het drietal (G, M, R) noemen we een V-context.*

Definitie 62. *[46] Zij (G, M, R) een V-context. De operatoren ${}^{\rightarrow_R} : V^G \rightarrow V^M$ en ${}^{\leftarrow_R} : V^M \rightarrow V^G$ worden gedefinieerd voor een V-map A van (G, μ) naar $(V, \rightarrow)$, B een V-map van (M, ν) naar $(V, \rightarrow)$, g in G en $m \in M$ als*

$$\begin{aligned} A^{\rightarrow_R}(m) &= \inf_{g \in G}(A(g) \rightarrow R(g, m)) \\ B^{\leftarrow_R}(g) &= \inf_{m \in M}(B(m) \rightarrow R(g, m)) \end{aligned}$$

Definitie 63 (V-concept). *Een V-concept in een V-context (G, M, R) is een koppel (A, B) met A een V-map van (G, μ) naar $(V, \rightarrow)$ en B een V-map van (M, ν) naar $(V, \rightarrow)$, waarvoor voldaan is aan $A^{\rightarrow} = B$ en $B^{\leftarrow} = A$.*

We zullen ons verder beperken tot de geresidueerde tralie $([0, 1], \leq, T, I_T, 1)$ met T een t-norm waarvan de partitiële afbeeldingen supmorfismen zijn en I_T de corresponderende residuele implicator. Een $[0, 1]$-context in $([0, 1], \leq, T, I, 1)$ wordt een vaagcontext genoemd, een $[0, 1]$-concept in $([0, 1], \leq, T, I, 1)$ wordt een vaagconcept genoemd. Meer expliciet hebben we:

Definitie 64 (Vaagcontext). *Een drietal (G, M, R), bestaande uit twee niet-ledige verzamelingen G en M en een vaagrelatie R van G naar M, wordt een vaagcontext genoemd.*

Definitie 65. *Zij (G, M, R) een vaagcontext. De operatoren ${}^{\rightarrow_R} : \mathcal{F}(G) \rightarrow \mathcal{F}(M)$ en ${}^{\leftarrow_R} : \mathcal{F}(M) \rightarrow \mathcal{F}(G)$ worden gedefinieerd voor een vaagverzameling A in G, een vaagverzameling B in M en m in M als*

$$\begin{aligned} A^{\rightarrow_R}(m) &= \inf_{g \in G} I(A(g), R(g, m)) \\ B^{\leftarrow_R}(m) &= \inf_{m \in M} I(B(m), R(g, m)) \end{aligned}$$

Ook hier zullen we, wanneer er geen gevaar is voor verwarring, de verwijzing naar R in ${}^{\rightarrow_R}$ en ${}^{\leftarrow_R}$ weglaten.

Definitie 66 (Vaagconcept). *Een vaagconcept in een vaagcontext (G, M, R) is een koppel (A, B) met A een vaagverzameling in G en B een vaagverzameling in M, waarvoor voldaan is aan $A^{\rightarrow} = B$ en $B^{\leftarrow} = A$. De klasse van alle vaagconcepten in de vaagcontext (G, M, R) noteren we als $\mathcal{B}(G, M, R)$.*

Eigenschap 15. *Zij (G, M, R) een vaagcontext, A_1 en A_2 vaagverzamelingen in G en B_1 en B_2 vaagverzamelingen in M, dan is steeds voldaan aan*

$$A_1 \subseteq A_2 \Rightarrow A_1^{\rightarrow} \supseteq A_2^{\rightarrow} \tag{3.28}$$

$$B_1 \subseteq B_2 \Rightarrow B_1^{\leftarrow} \supseteq B_2^{\leftarrow} \tag{3.29}$$

Bewijs. We tonen als voorbeeld (3.28) aan. Het bewijs van (3.29) is volledig analoog. Onderstel dus $A_1 \subseteq A_2$, m.a.w. $(\forall g \in G)(A_1(g) \leq A_2(g))$. We hebben voor willekeurige $m \in M$

$$(A_1^{\rightarrow})(m) = \inf_{g \in G} I(A_1(g), R(g, m))$$

Gezien de onderstelling en het dalend karakter van de eerste partiële afbeeldingen van I hebben we

$$\inf_{g\in G} I(A_1(g), R(g,m)) \geq \inf_{g\in G} I(A_2(g), R(g,m)) = A_2^{\twoheadrightarrow}(m)$$

en dus het gestelde. □

Voor twee vaagconcepten (A_1, B_1) en (A_2, B_2) hebben we bijgevolg:

$$A_1 \subseteq A_2 \Leftrightarrow B_1 \supseteq B_2 \tag{3.30}$$

en bovendien

$$A_1 = A_2 \Leftrightarrow B_1 = B_2 \tag{3.31}$$

aangezien voor A en B willekeurige vaagverzamelingen in een universum X geldt

$$\begin{aligned} A \subseteq B \text{ en } B \subseteq A &\Leftrightarrow (\forall x \in X)(A(x) \leq B(x)) \text{ en } (\forall x \in X)(B(x) \leq A(x)) \\ &\Leftrightarrow (\forall x \in X)(A(x) \leq B(x) \text{ en } B(x) \leq A(x)) \\ &\Leftrightarrow (\forall x \in X)(A(x) = B(x)) \\ &\Leftrightarrow A = B \end{aligned}$$

Eigenschap 16. *Zij (G, M, R) een vaagcontext, $(A_i)_{i\in I}$ een familie vaagverzamelingen in G en $(B_i)_{i\in I}$ een familie vaagverzamelingen in M, dan is steeds voldaan aan*

$$(\bigcap_{i\in I} A_i)^{\twoheadrightarrow} \supseteq \bigcup_{i\in I} A_i^{\twoheadrightarrow} \tag{3.32}$$

$$(\bigcup_{i\in I} A_i)^{\twoheadrightarrow} = \bigcap_{i\in I} A_i^{\twoheadrightarrow} \tag{3.33}$$

$$(\bigcup_{i\in I} B_i)^{\twoheadleftarrow} = \bigcap_{i\in I} B_i^{\twoheadleftarrow} \tag{3.34}$$

$$(\bigcap_{i\in I} B_i)^{\twoheadleftarrow} \supseteq \bigcup_{i\in I} B_i^{\twoheadleftarrow} \tag{3.35}$$

Bewijs.

- We tonen eerst (3.32) aan. Voor elke i_0 in I hebben we wegens (3.14)

$$\bigcap_{i\in I} A_i \subseteq A_{i_0}$$

en dus wegens (3.28)

$$(\bigcap_{i\in I} A_i)^{\twoheadrightarrow} \supseteq A_{i_0}^{\twoheadrightarrow}$$

Wegens (3.17) volgt hieruit

$$(\bigcap_{i\in I} A_i)^{\twoheadrightarrow} \supseteq \bigcup_{i\in I} A_i^{\twoheadrightarrow}$$

en dus het gestelde

- Voor (3.33) krijgen we wegens de definitie van $^{\twoheadrightarrow}$ en $\bigcup$ voor m in M

$$(\bigcup_{i\in I} A_i)^{\twoheadrightarrow}(m) = \inf_{g\in G} I(\sup_{i\in I} A_i(g), R(g,m))$$

Uit eig. 8 volgt dan

$$\begin{aligned}(\bigcup_{i\in I} A_i)^{\twoheadrightarrow}(m) &= \inf_{g\in G}\inf_{i\in I} I(A_i(g), R(g,m)) \\ &= \inf_{i\in I}\inf_{g\in G} I(A_i(g), R(g,m)) \\ &= (\bigcap_{i\in I} A_i^{\twoheadrightarrow})(m)\end{aligned}$$

- Het bewijs van (3.34) is volledig analoog aan het bewijs van (3.33)
- Het bewijs van (3.35) is volledig analoog aan het bewijs van (3.32)

□

Eigenschap 17. *Zij $A \in \mathcal{F}(G)$ en $B \in \mathcal{F}(M)$, dan geldt er*

$$A^{\twoheadrightarrow\twoheadleftarrow\twoheadrightarrow} = A^{\twoheadrightarrow} \tag{3.36}$$

$$B^{\twoheadleftarrow\twoheadrightarrow\twoheadleftarrow} = B^{\twoheadleftarrow} \tag{3.37}$$

Bewijs. We tonen als voorbeeld (3.36) aan. Het bewijs van (3.37) is volledig analoog. We bewijzen eerst $A^{\twoheadrightarrow\twoheadleftarrow\twoheadrightarrow} \subseteq A^{\twoheadrightarrow}$. Aangezien het infimum een ondergrens is geldt

$$\Big(\forall g \in G\Big)\Big(\forall m' \in M\Big)\Big(\inf_{g'\in G} I\big(A(g'), R(g',m')\big) \leq I\big(A(g), R(g,m')\big)\Big)$$

omdat de eerste partiële afbeeldingen van een implicator dalend zijn volgt hieruit

$$\Big(\forall g \in G\Big)\Big(\forall m' \in M\Big)\Big(I\big(\inf_{g'\in G} I\big(A(g'), R(g',m')\big), R(g,m')\big) \geq I\big(I\big(A(g), R(g,m')\big), R(g,m')\big)\Big)$$

waaruit wegens (3.9) volgt

$$\Big(\forall g \in G\Big)\Big(\forall m' \in M\Big)\Big(I\big(\inf_{g'\in G} I\big(A(g'), R(g',m')\big), R(g,m')\big) \geq A(g)\Big)$$

wegens de monotoniciteit van het infimum volgt hieruit

$$\Big(\forall g \in G\Big)\Big(\inf_{m'\in M} I\big(\inf_{g'\in G} I\big(A(g'), R(g',m')\big), R(g,m')\big) \geq \inf_{m'\in M} A(g) = A(g)\Big)$$

zodat opnieuw wegens het dalend zijn van de eerste partiële afbeeldingen van I voor willekeurige m in M volgt

$$\Big(\forall g \in G\Big)\Big(I\big(\inf_{m'\in M} I\big(\inf_{g'\in G} I\big(A(g'), R(g',m')\big), R(g,m')\big), R(g,m)\big) \leq I\big(A(g), R(g,m)\big)\Big)$$

hieruit volgt ten slotte

$$\inf_{g\in G} I\big(\inf_{m'\in M} I\big(\inf_{g'\in G} I\big(A(g'), R(g',m')\big), R(g,m')\big), R(g,m)\big) \leq \inf_{g\in G} I\big(A(g), R(g,m)\big)$$

m.a.w. $A^{\twoheadrightarrow\twoheadleftarrow\twoheadrightarrow}(m) \leq A^{\twoheadrightarrow}(m)$, voor willekeurige m in M. Omgekeerd geldt voor m in M wegens (3.9) dat

$$\Big(\forall g' \in G\Big)\Big(\inf_{g\in G} I(A(g), R(g,m)) \leq I\big(I\big(\inf_{g\in G} I\big(A(g), R(g,m)\big), R(g',m)\big), R(g',m)\big)\Big)$$

wegens het dalend zijn van de eerste partiële afbeelding van I en het feit dat het infimum een ondergrens is, volgt er

$$\Big(\forall g' \in G\Big)\Big(\inf_{g\in G} I\big(A(g), R(g,m)\big) \leq I\big(\inf_{m'\in M} I\big(\inf_{g\in G} I\big(A(g), R(g,m')\big), R(g',m')\big), R(g',m)\big)\Big)$$

aangezien het infimum de grootste ondergrens is, kunnen we besluiten

$$\inf_{g\in G} I\big(A(g), R(g,m)\big) \leq \inf_{g'\in G} I\big(\inf_{m'\in M} I\big(\inf_{g\in G} I\big(A(g), R(g,m')\big), R(g',m')\big), R(g',m)\big)$$

m.a.w. $A^{\twoheadrightarrow}(m) \leq A^{\twoheadrightarrow\twoheadleftarrow\twoheadrightarrow}(m)$, voor willekeurige m in M en dus het gestelde $\square$

Stelling 2. *Zij (G, M, R) een vaagcontext. De structuur $(\mathcal{B}(G, M, R), \vee, \wedge)$ is een complete tralie waarbij de join- en meet-bewerking respectievelijk gedefinieerd worden voor een familie $(A_i, B_i)_{i\in I}$ van vaagconcepten in (G, M, R) als*

$$\sup_{i\in I}(A_i, B_i) = ((\bigcup_{i\in I} A_i)^{\twoheadrightarrow\twoheadleftarrow}, \bigcap_{i\in I} B_i)$$
$$\inf_{i\in I}(A_i, B_i) = (\bigcap_{i\in I} A_i, (\bigcup_{i\in I} B_i)^{\twoheadleftarrow\twoheadrightarrow})$$

De met deze tralie corresponderende ordestructuur wordt voor vaagconcepten (A, B) en (C, D) in $\mathcal{B}(G, M, R)$ gegeven door

$$(A, B) \leq (C, D) \Leftrightarrow A \subseteq C \Leftrightarrow B \supseteq D$$

Bewijs. Definiëren we de binaire relatie $\leq$ voor vaagconcepten (A, B) en (C, D) in $\mathcal{B}(G, M, R)$ als

$$(A, B) \leq (C, D) \Leftrightarrow A \subseteq C$$

Wegens (3.30) hebben we dus ook

$$(A, B) \leq (C, D) \Leftrightarrow D \subseteq B$$

We tonen nu eerst aan dat $(\mathcal{B}(G, M, R), \leq)$ een poset is. Voor de reflexiviteit krijgen we voor elke (A, B) in $\mathcal{B}(G, M, R)$

$$\begin{aligned}(A, B) \leq (A, B) &\Leftrightarrow A \subseteq A \\ &\Leftrightarrow (\forall g \in G)(A(g) \leq A(g))\end{aligned}$$

Hieraan is steeds voldaan. Voor de antisymmetrie krijgen we voor willekeurige (A_1, B_1) en (A_2, B_2) in $\mathcal{B}(G, M, R)$

$$\begin{aligned}&(A_1, B_1) \leq (A_2, B_2) \text{ en } (A_2, B_2) \leq (A_1, B_1) \\ &\quad\Leftrightarrow A_1 \subseteq A_2 \text{ en } A_2 \subseteq A_1 \\ &\quad\Leftrightarrow (\forall g \in G)(A_1(g) \leq A_2(g)) \text{ en } (\forall g \in G)(A_2(g) \leq A_1(g)) \\ &\quad\Leftrightarrow (\forall g \in G)(A_1(g) \leq A_2(g) \text{ en } A_2(g) \leq A_1(g)) \\ &\quad\Rightarrow (\forall g \in G)(A_1(g) = A_2(g)) \\ &\quad\Leftrightarrow A_1 = A_2\end{aligned}$$

Wegens (3.31) volgt hieruit $(A_1, B_1) = (A_2, B_2)$. Voor de transitiviteit ten slotte, krijgen we voor willekeurige (A_1, B_1), (A_2, B_2) en (A_3, B_3)

$$\begin{aligned}
&(A_1, B_1) \leq (A_2, B_2) \text{ en } (A_2, B_2) \leq (A_3, B_3)\\
&\quad \Leftrightarrow A_1 \subseteq A_2 \text{ en } A_2 \subseteq A_3\\
&\quad \Leftrightarrow (\forall g \in G)(A_1(g) \leq A_2(g)) \text{ en } (\forall g \in G)(A_2(g) \leq A_3(g))\\
&\quad \Leftrightarrow (\forall g \in G)(A_1(g) \leq A_2(g) \text{ en } A_2(g) \leq A_3(g))\\
&\quad \Rightarrow (\forall g \in G)(A_1(g) \leq A_3(g))\\
&\quad \Leftrightarrow A_1 \subseteq A_3\\
&\quad \Leftrightarrow (A_1, B_1) \subseteq (A_3, B_3)
\end{aligned}$$

Om te bewijzen dat $(\mathcal{B}(G, M, R), \leq)$ een complete tralie is, moeten we aantonen dat voor een willekeurige familie $(A_i, B_i)_{i \in I}$ van vaagconcepten in $\mathcal{B}(G, M, R)$ het supremum en infimum bestaat. We tonen aan dat $\sup_{i \in I}(A_i, B_i) = ((\bigcup_{i \in I} A_i)^{\twoheadrightarrow\twoheadleftarrow}, \bigcap_{i \in I} B_i)$. Het bewijs voor het infimum is analoog. We verifiëren vooreerst dat $((\bigcup_{i \in I} A_i)^{\twoheadrightarrow\twoheadleftarrow}, \bigcap_{i \in I} B_i)$ wel degelijk een vaagconcept is. Uit (3.36), (3.33) en $A_i^{\twoheadrightarrow} = B_i$ voor elke i in I volgt achtereenvolgens

$$(\bigcup_{i \in I} A_i)^{\twoheadrightarrow\twoheadleftarrow\twoheadrightarrow} = (\bigcup_{i \in I} A_i)^{\twoheadrightarrow} = \bigcap_{i \in I} A_i^{\twoheadrightarrow} = \bigcap_{i \in I} B_i$$

Uit $A_i^{\twoheadrightarrow} = B_i$ voor elke i in I en (3.33) volgt achtereenvolgens

$$(\bigcap_{i \in I} B_i)^{\twoheadleftarrow} = (\bigcap_{i \in I} A_i^{\twoheadrightarrow})^{\twoheadleftarrow} = (\bigcup_{i \in I} A_i)^{\twoheadrightarrow\twoheadleftarrow}$$

Bijgevolg is $((\bigcup_{i \in I} A_i)^{\twoheadrightarrow\twoheadleftarrow}, \bigcap_{i \in I} B_i)$ een vaagconcept. We tonen vervolgens aan dat deze uitdrukking bovendien een bovengrens is voor $(A_i, B_i)_{i \in I}$. Voor willekeurige i in I hebben we

$$(A_i, B_i) \leq ((\bigcup_{i \in I} A_i)^{\twoheadrightarrow\twoheadleftarrow}, \bigcap_{i \in I} B_i) \Leftrightarrow B_i \supseteq \bigcap_{i \in I} B_i$$

waaraan wegens (3.14) steeds voldaan is. We tonen ten slotte aan dat $((\bigcup_{i \in I} A_i)^{\twoheadrightarrow\twoheadleftarrow}, \bigcap_{i \in I} B_i)$ de kleinste bovengrens is. We moeten m.a.w. aantonen dat voor een willekeurig concept (C, D) in $\mathcal{B}(G, M, R)$ geldt

$$(\forall i \in I)((A_i, B_i) \leq (C, D)) \Rightarrow ((\bigcup_{i \in I} A_i)^{\twoheadrightarrow\twoheadleftarrow}, \bigcap_{i \in I} B_i) \leq (C, D))$$

Onderstellen we nu

$$(\forall i \in I)((A_i, B_i) \leq (C, D))$$

of equivalent

$$(\forall i \in I)(D \subseteq B_i)$$

wegens (3.16) volgt hieruit

$$D \subseteq \bigcap_{i \in I} B_i$$

wat equivalent is met

$$((\bigcup_{i \in I} A_i)^{\twoheadrightarrow\twoheadleftarrow}, \bigcap_{i \in I} B_i) \leq (C, D)$$

en dus het gestelde. □

3.6.3 Boven- en onderbenaderingen

We beschouwen eerst het scherpe geval. In [47] wordt het begrip ruwconcept analyse (Eng. rough concept analysis) geïntroduceerd. Boven- en onderbenaderingen van concepten worden gedefinieerd a.d.h.v. boven- en onderbenadering van de (formele) context waarin gewerkt wordt.

Definitie 67. *[47] Zij (G, M, R) een (formele) context en (G, E) een benaderingsruimte. De boven- en onderbenadering van (G, M, R) worden respectievelijk gegeven door $(G, M, \overline{E}R)$ en $(G, M, \underline{E}R)$. Hierbij worden $\overline{E}R$ en $\underline{E}R$ voor g in G en m in M gedefinieerd door*

$$\begin{aligned}(g,m) \in \overline{E}R &\Leftrightarrow g \in \overline{E}(Rm) \Leftrightarrow Eg \cap Rm \neq \emptyset \\ (g,m) \in \underline{E}R &\Leftrightarrow g \in \underline{E}(Rm) \Leftrightarrow Eg \subseteq Rm\end{aligned}$$

De boven- of onderbenadering van een concept in de context (G, M, R) m.b.t. de benaderingsruimte (G, E) worden nu bepaald door met dit concept een concept te associëren in de boven- of onderbenadering van (G, M, R).

Definitie 68. *[47] Zij (A, B) een concept in de context (G, M, R) en (G, E) een benaderingsruimte. De bovenbenadering $\overline{E}(A, B)$ en onderbenadering $\underline{E}(A, B)$ van (A, B) worden respectievelijk gedefinieerd door*

$$\begin{aligned}\overline{E}(A,B) &= (B^{\twoheadleftarrow_{\overline{E}R}}, B^{\twoheadleftarrow_{\overline{E}R}\twoheadrightarrow_{\overline{E}R}}) \\ \underline{E}(A,B) &= (B^{\twoheadleftarrow_{\underline{E}R}}, B^{\twoheadleftarrow_{\underline{E}R}\twoheadrightarrow_{\underline{E}R}})\end{aligned}$$

Uit eigenschap 17 volgt onmiddellijk dat de gegeven definities wel degelijk concepten voorstellen. We beschouwen nu, op een volledig analoge manier, het geval waarbij de verzameling attributen voorzien is van een ononderscheidbaarheidsrelatie.

Definitie 69. *Zij (G, M, R) een (formele) context en (M, E) een benaderingsruimte. De boven- en onderbenadering van (G, M, R) worden respectievelijk gegeven door $(G, M, \overline{E}R)$ en $(G, M, \underline{E}R)$. Hierbij worden $\overline{E}R$ en $\underline{E}R$ voor g in G en m in M gedefinieerd door*

$$\begin{aligned}(g,m) \in \overline{E}R &\Leftrightarrow m \in \overline{E}(gR) \Leftrightarrow Em \cap gR \neq \emptyset \\ (g,m) \in \underline{E}R &\Leftrightarrow m \in \underline{E}(gR) \Leftrightarrow Em \subseteq gR\end{aligned}$$

Definitie 70. *Zij (A, B) een concept in de context (G, M, R) en (M, E) een benaderingsruimte. De bovenbenadering $\overline{E}(A, B)$ en onderbenadering $\underline{E}(A, B)$ van (A, B) worden respectievelijk gedefinieerd door*

$$\begin{aligned}\overline{E}(A,B) &= (A^{\twoheadrightarrow_{\overline{E}R}\twoheadleftarrow_{\overline{E}R}}, A^{\twoheadrightarrow_{\overline{E}R}}) \\ \underline{E}(A,B) &= (A^{\twoheadrightarrow_{\underline{E}R}\twoheadleftarrow_{\underline{E}R}}, A^{\twoheadrightarrow_{\underline{E}R}})\end{aligned}$$

Uit eigenschap 17 volgt onmiddellijk dat de gegeven definities wel degelijk concepten voorstellen. Uitbreiding naar vaagconcepten en (veralgemeende) vaagbenaderingsruimten ligt nu voor de hand.

Definitie 71. *Zij (G, M, R) een vaagcontext en (G, E) een (veralgemeende) vaagbenaderingsruimte. De boven- en onderbenadering van (G, M, R) worden respectievelijk gegeven door*

$(G, M, \overline{E}R)$ *en* $(G, M, \underline{E}R)$*. Hierbij worden* $\overline{E}R$ *en* $\underline{E}R$ *voor* g *in* G *en* m *in* M *gedefinieerd door*

$$\begin{aligned}\overline{E}R(g,m) &= \overline{E}(Rm)(g) = \sup_{g' \in G} T(R(g',m), E(g,g')) = (E \circ_T) R(g,m) \\ \underline{E}R(g,m) &= \underline{E}(Rm)(g) = \inf_{g' \in G} I(E(g,g'), R(g',m)) = (E \triangleleft_I) R(g,m)\end{aligned}$$

Definitie 72. *Zij* (A, B) *een vaagconcept in de vaagcontext* (G, M, R) *en* (G, E) *een (veralgemeende) vaagbenaderingsruimte. De bovenbenadering* $\overline{E}(A, B)$ *en onderbenadering* $\underline{E}(A, B)$ *van* (A, B) *worden respectievelijk gedefinieerd door*

$$\begin{aligned}\overline{E}(A,B) &= (B^{\overline{\twoheadleftarrow}_{\overline{E}R}}, B^{\overline{\twoheadleftarrow}_{\overline{E}R}\overline{\twoheadrightarrow}_{\overline{E}R}}) \\ \underline{E}(A,B) &= (B^{\underline{\twoheadleftarrow}_{\underline{E}R}}, B^{\underline{\twoheadleftarrow}_{\underline{E}R}\underline{\twoheadrightarrow}_{\underline{E}R}})\end{aligned}$$

Definitie 73. *Zij* (G, M, R) *een vaagcontext en* (M, E) *een (veralgemeende) vaagbenaderingsruimte. De boven- en onderbenadering van* (G, M, R) *worden respectievelijk gegeven door* $(G, M, \overline{E}R)$ *en* $(G, M, \underline{E}R)$*. Hierbij worden* $\overline{E}R$ *en* $\underline{E}R$ *voor* g *in* G *en* m *in* M *gedefinieerd door*

$$\begin{aligned}\overline{E}R(g,m) &= \overline{E}(gR)(m) = \sup_{m' \in M} T(R(g,m'), E(m,m')) = (E \circ_T R^{-1})(g,m) \\ \underline{E}R(g,m) &= \underline{E}(gR)(m) = \inf_{m' \in M} I(E(m,m'), R(g,m')) = (E \triangleleft_I R^{-1}(g,m)\end{aligned}$$

Definitie 74. *Zij* (A, B) *een vaagconcept in de vaagcontext* (G, M, R) *en* (M, E) *een (veralgemeende) vaagbenaderingsruimte. De bovenbenadering* $\overline{E}(A, B)$ *en onderbenadering* $\underline{E}(A, B)$ *van* (A, B) *worden respectievelijk gedefinieerd door*

$$\begin{aligned}\overline{E}(A,B) &= (A^{\overline{\twoheadrightarrow}_{\overline{E}R}\overline{\twoheadleftarrow}_{\overline{E}R}}, A^{\overline{\twoheadrightarrow}_{\overline{E}R}}) \\ \underline{E}(A,B) &= (A^{\underline{\twoheadrightarrow}_{\underline{E}R}\underline{\twoheadleftarrow}_{\underline{E}R}}, A^{\underline{\twoheadrightarrow}_{\underline{E}R}})\end{aligned}$$

Hoofdstuk 4

Vaagmieren

4.1 Inleiding

We introduceren in dit hoofdstuk het gebruik van vaagregels voor een zelf ontworpen miergebaseerd clusteringsalgoritme. We zullen ons hierbij baseren op het algoritme van Monmarché, maar niet zonder enkele belangrijke wijzigingen:

- We zullen ons concentreren op een niet-parallelle implementatie. Zoals we reeds in hoofdstuk 2 stelden, heeft het gebruik van meerdere mieren weinig zin wanneer we slechts over één processor beschikken. We beschouwen daarom in dit hoofdstuk slechts één mier. De wijzigingen die we doorvoeren, sluiten het gebruik van meerdere mieren echter niet uit.

- Het gebruik van een roostervoorstelling heeft als belangrijk nadeel dat mieren heel wat tijd spenderen aan het zoeken naar clusters. Wanneer, zoals bij het algoritme van Monmarché, verschillende objecten op hetzelfde vakje toegelaten worden, wordt het aantal vrije vakjes steeds groter. Dit heeft als gevolg dat mieren hoe langer hoe meer tijd verliezen. Regelmatig herschalen van het rooster zou een uitkomst kunnen bieden, maar brengt een hoge kost met zich mee en is in elk geval geen elegante oplossing. Ook de meeste varianten op het algoritme van Lumer en Faieta proberen vooral de tijd die mieren spenderen aan het zoeken naar clusters op het rooster te verminderen. De impact van deze verbeteringen is meestal onduidelijk en de computationele kost die ze met zich meebrengen groot. We zullen daarom deze roostervoorstelling volledig weglaten.

- Het gebruik van verschillende fasen werd door Monmarché geïntroduceerd om zowel mieren met een oneindige capaciteit te kunnen gebruiken als mieren met capaciteit 1. Mieren met capaciteit 1 zorgen voor een groot aantal hechte, zuivere clusters. Mieren met oneindige capaciteit kunnen deze hechte clusters samennemen om een kleiner aantal grote clusters te bekomen. De efficiëntie wordt drastisch verhoogd door het gebruik van mieren met oneindige capaciteit. Het ontbreken van mieren met capaciteit 1 in de tweede fase, heeft echter een significante daling van de zuiverheid van de clusters als gevolg. Hierdoor is een combinatie met het k-gemiddelden algoritme noodzakelijk. Wanneer we nu regels voor het opnemen zouden kunnen vinden waarbij een mier een hele hoop opneemt als deze hoop voldoende zuiver is (volgens een bepaald criterium) en een individueel object opneemt in het andere geval, kunnen we zowel het gebruik van

verschillende fasen als de combinatie met k-gemiddelden elimineren. De flexibiliteit die een beschrijving a.d.h.v. vaagregels met zich meebrengt, maakt dit mogelijk.

4.2 Een vaagmieralgoritme

4.2.1 Enkele definities

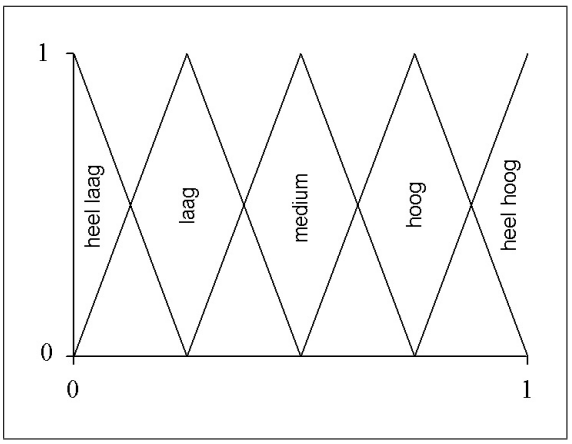
Noemen we X de te clusteren gegevensverzameling en E een vaagrelatie in X die reflexief en T_W-transitief is. Hierbij is T_W de Łukasiewicz t-norm. Voor elementen x en y in X, geeft $E(x, y)$ de mate van similariteit aan tussen x en y. Merk op dat we hierbij niet eisen dat E symmetrisch is. We zullen hier gebruik van maken in het laatste hoofdstuk om zoekresultaten te clusteren. Een hoop is steeds een niet lege deelverzameling van X en we veronderstellen dat we op een of andere manier het centrum van een hoop kunnen bepalen. Een eerste mogelijkheid is om als centrum van een hoop het object van die hoop te kiezen dat het meest representatief is volgens een zeker criterium. We zullen deze mogelijkheid benutten in het laatste hoofdstuk. Wanneer de objecten als vectoren beschreven zijn, kunnen we ook het zwaartepunt van deze vectoren als centrum kiezen. Het centrum van een hoop is in dit geval niet noodzakelijk een object van die hoop. Deze tweede mogelijkheid zullen we illustreren in paragraaf 4.3. Zij H een hoop met centrum c, dan noteren we $avg(H)$ voor de gemiddelde similariteit tussen elementen van H en het centrum c, m.a.w.

$$avg(H) = \frac{1}{|H|} \sum_{h \in H} E(h, c)$$

waarbij steeds voldaan is aan $|H| > 0$ aangezien we een hoop als een niet lege deelverzameling hebben gedefinieerd. We noteren $min(H)$ voor de minimale similariteit tussen elementen van H en het centrum c, m.a.w.

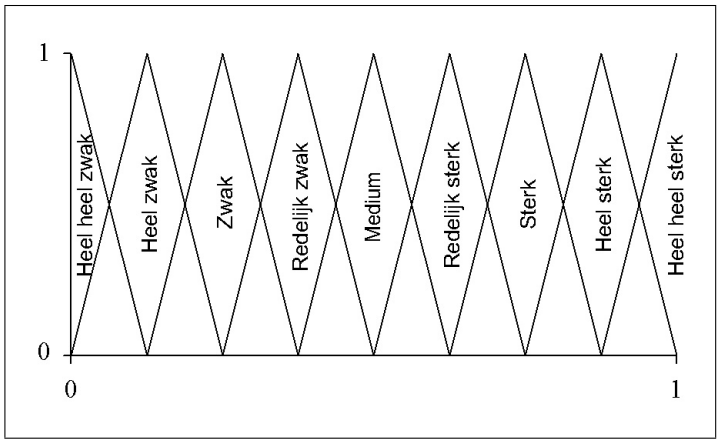
$$min(H) = \min_{h \in H} E(h, c)$$

Onze mier zal deze gemiddelde en minimale similariteit ervaren als `Heel laag`, `Laag`, `Medium`, `Hoog` of `Heel hoog`. Deze linguïstische termen worden gerepresenteerd d.m.v. de vaagverzamelingen in $[0, 1]$ uit figuur 4.1. Bij de beschrijving van het algoritme, baseren we ons op het



Figuur 4.1: Linguïstische termen voor de similariteit

model voor taakverdeling van Bonabeau et al. uit paragraaf 1.3. Onze mier zal de stimulus voor een bepaalde taak ervaren als zijnde `Heel Heel Zwak`, `Heel Zwak`, `Zwak`, `Redelijk Zwak`, `Medium`, `Redelijk Sterk`, `Sterk`, `Heel Sterk` of `Heel Heel Sterk`. Deze linguïstische termen worden gerepresenteerd d.m.v. de vaagverzamelingen in $[0, 1]$ uit figuur 4.2.



Figuur 4.2: Linguïstische termen voor de stimulus

4.2.2 Opnemen van objecten en hopen

Het opnemen van één enkel object en het opnemen van een hele hoop kunnen we beschouwen als twee verschillende taken, elk met hun eigen stimulus en drempelwaarde. Noemen we s_{obj} en θ_{obj} respectievelijk de stimulus en drempelwaarde geassocieerd met het opnemen van een object en s_{hoop} en θ_{hoop} respectievelijk de stimulus en drempelwaarde geassocieerd met het opnemen van een volledige hoop. We definiëren de probabiliteit P_{obj} dat een object wordt opgenomen en de probabiliteit P_{hoop} dat een volledige hoop wordt opgenomen dan als

$$P_{obj} = \frac{s_{obj}}{s_{obj} + s_{hoop}} \cdot \frac{s_{obj}^{m_1}}{\theta_{obj}^{m_1} + s_{obj}^{m_1}} \tag{4.1}$$

$$P_{hoop} = \frac{s_{hoop}}{s_{obj} + s_{hoop}} \cdot \frac{s_{hoop}^{m_2}}{\theta_{hoop}^{m_2} + s_{hoop}^{m_2}} \tag{4.2}$$

Hierbij zijn m_1 en m_2 strikt positieve, natuurlijke getallen. De drempelwaarden θ_{obj} en θ_{hoop} zijn reële constanten uit $[0, 1]$. De stimuli s_{obj} en s_{hoop} nemen reële waarden aan in $[0, 1]$ en worden bepaald door het evalueren van vaagregels.

We willen dat de mier een hele hoop opneemt in het geval deze hoop zuiver is. Hierbij beschouwen we een hoop H zuiver, als $min(H)$ ongeveer gelijk is aan $avg(H)$ en als $min(H)$ bovendien niet al te klein is. Noemen we nu $A = avg(H)$ en $M = min(H)$. De stimulus voor het opnemen van de volledige hoop H, wordt dan bepaald door het evalueren van de vaagregels die samengevat worden in tabel 4.1. Aangezien de gemiddelde similariteit steeds hoger zal zijn dan de minimale similariteit, worden de elementen boven de hoofddiagonaal in deze tabel niet beschouwd. De eerste regel moet bijvoorbeeld geïnterpreteerd worden als

Als M `Heel Hoog` is en A `Heel Hoog` is dan is s_{hoop} `Heel Heel Sterk`

	A is H. Hoog	A is Hoog	A is Medium	A is Laag	A is H. Laag
M is H. Hoog	H. H. Sterk	-	-	-	-
M is Hoog	Medium	H. Sterk	-	-	-
M is Medium	Zwak	R. Zwak	Sterk	-	-
M is Laag	H.H. Zwak	H. Zwak	Zwak	R. Sterk	-
M is H. Laag	H.H. Zwak	H.H. Zwak	H.H. Zwak	H. Zwak	Medium

Tabel 4.1: Regels voor het opnemen van een volledige hoop

Aangezien we hier in feite een onbekende functie benaderen, hebben we te maken met een toepassing van het conjunctiegebaseerd model. We maken gebruik van het minimum voor het modelleren van zowel de vaagregels zelf als de conjunctie in het antecedent van de regels. We gebruiken het maximum als aggregatie-operator. Het resultaat van de evaluatie wordt omgevormd tot een scherpe waarde door gebruik te maken van de zwaartepuntmethode.

We willen dat de mier een individueel object opneemt van een hoop H wanneer de minimale similariteit $min(H)$ veel kleiner is dan de gemiddelde similariteit $avg(H)$. Wanneer een individueel object moet worden opgenomen, kiest de mier steeds het object waarvan de similariteit met het centrum van de hoop minimaal is. De stimulus voor het opnemen van een individueel object uit de hoop H, wordt bepaald door het evalueren van de vaagregels die samengevat worden in tabel 4.2. We merken hierbij op dat we bij het opnemen van een individueel object heel wat toleranter kunnen zijn dan bij het opnemen van een volledige hoop. Wanneer een hoop wordt opgenomen die niet zuiver is, kan dit tot een foutief resultaat leiden. Karakteristieken van een niet zuivere hoop H, zoals $avg(H)$ en $min(H)$, kunnen immers in grote mate beïnvloed zijn door objecten van de hoop die er eigenlijk niet in thuishoren.

	A is H. Hoog	A is Hoog	A is Medium	A is Laag	A is H. Laag
M is H. Hoog	Medium	-	-	-	-
M is Hoog	Sterk	R. Sterk	-	-	-
M is Medium	H.H. Sterk	H. Sterk	Sterk	-	-
M is Laag	H.H. Sterk	H.H. Sterk	H.H. Sterk	H. Sterk	-
M is Heel Laag	H.H. Sterk	H.H. Sterk	H.H. Sterk	H.H. Sterk	H.H. Sterk

Tabel 4.2: Regels voor het opnemen van een individueel object

4.2.3 Neerleggen van objecten en hopen

Wanneer de mier reeds een hoop, eventueel enkel bestaande uit een individueel object, aan het dragen is, is de enige taak die de mier kan uitvoeren het neerleggen van deze hoop. Zij s_{neer} en θ_{neer} respectievelijk de stimulus en drempelwaarde geassocieerd met deze taak. We

definiëren de probabiliteit dat de hoop wordt neergelegd als

$$P_{neer} = \frac{s_{neer}^{n_i}}{\theta_{neer}^{n_i} + s_{neer}^{n_i}} \tag{4.3}$$

Hierbij is $i \in \{1, 2\}$ en zijn n_1 en n_2 strikt positieve, natuurlijke getallen. Wanneer de mier een individueel object aan het dragen is, wordt n_1 gebruikt; wanneer de mier een volledige hoop, die uit meer dan 1 object bestaat, aan het dragen is, wordt n_2 gebruikt. De drempelwaarde θ_{neer} is een reële constante uit $[0, 1]$. De stimulus s_{neer} neemt reële waarden aan in $[0, 1]$ en wordt bepaald door het evalueren van vaagregels.

Veronderstellen we nu dat de mier een hoop L aan het dragen is en overweegt om L op de hoop H neer te leggen. Dit is gewenst als de gemiddelde similariteit tussen de elementen van L en het centrum van H niet (veel) kleiner is dan $avg(H)$. Aangezien de mier enkel zuivere hopen opneemt[1], zal de minimale similariteit van $L \cup H$ dan ook niet (veel) kleiner zijn dan $min(H)$. Het is omwille van efficiëntieredenen niet gewenst om de gemiddelde similariteit tussen de objecten van L en het centrum van H te berekenen voor elke hoop H die in overweging wordt genomen. Aangezien $avg(L)$ reeds bepaald werd voor de evaluatie van de vaagregels bij het opnemen, kunnen we hiervan gebruik maken zonder bijkomende kost. We zullen daarom $avg(H)$ vergelijken met $B = T_W(E(c_L, c_H), avg(L))$. Hierbij is c_L het centrum van L en c_H het centrum van H. Wegens de onderstelde T_W-transitiviteit van E is B een ondergrens voor de gemiddelde similariteit tussen elementen van L en c_H, we hebben immers

$$\begin{aligned}
T_W(E(c_L, c_H), avg(L)) &= T_W\left(E(c_L, c_H), \frac{1}{|L|}\sum_{l \in L} E(l, c_L)\right) \\
&= \max\left(0, E(c_L, c_H) + \frac{1}{|L|}\sum_{l \in L} E(l, c_L) - 1\right) \\
&= \max\left(0, \frac{1}{|L|}\sum_{l \in L}(E(c_L, c_H) + E(l, c_L) - 1)\right) \\
&\leq \frac{1}{|L|}\sum_{l \in L} \max\left(0, E(c_L, c_H) + E(l, c_L) - 1\right) \\
&= \frac{1}{|L|}\sum_{l \in L} T_W(E(c_L, c_H), E(l, c_L)) \\
&\leq \frac{1}{|L|}\sum_{l \in L} E(l, c_H)
\end{aligned}$$

Tabel 4.3 geeft een overzicht van de vaagregels die geëvalueerd moeten worden om de stimulus s_{neer} te bepalen.

4.2.4 Het algoritme

We houden tijdens de uitvoering van het algoritme een lijst van hopen bij. Initieel is er een hoop voor elk object uit de gegevensverzameling X. Een hoop opnemen komt dan overeen

[1]Een individueel object kunnen we ook beschouwen als een (perfect) zuivere hoop aangezien uit $|H| = 1$ volgt dat $avg(H) = min(H)$

	A is H. Hoog	A is Hoog	A is Medium	A is Laag	A is H. Laag
B is H. Hoog	R. Sterk	Sterk	H. Sterk	H.H. Sterk	H.H. Sterk
B is Hoog	Zwak	R. Sterk	Sterk	H. Sterk	H.H. Sterk
B is Medium	H.H. Zwak	Zwak	R. Sterk	Sterk	H. Sterk
B is Laag	H.H. Zwak	H.H. Zwak	Zwak	R. Sterk	Sterk
B is H. Laag	H.H. Zwak	H.H. Zwak	H.H. Zwak	Zwak	R. Sterk

Tabel 4.3: Regels voor het neerleggen van een hoop of een object

met een hoop verwijderen uit de lijst. In elke iteratie kiest de mier willekeurig een hoop H uit deze lijst en voert de volgende stappen uit:

- Als de mier nog vrij is
  - Als H slechts één object bevat, wordt dit object met een vaste probabiliteit opgenomen. We kunnen deze probabiliteit bijvoorbeeld gelijk stellen aan 1, een individueel object wordt dan m.a.w. steeds opgenomen.
  - Afhankelijk van de manier waarop het centrum van een hoop gedefinieerd wordt, zou het kunnen dat het vergelijken van de minimale similariteit met de gemiddelde similariteit van een hoop met grootte 2 niet zinvol is. Zij het centrum c van de hoop $H = \{a, b\}$ bijvoorbeeld zo gedefinieerd dat $E(a, c) = E(b, c)$. We hebben dan dat $avg(H) = min(H)$. We beschouwen daarom dit geval afzonderlijk. Als H nu bestaat uit twee objecten a en b, wordt één ervan opgenomen met probabiliteit$(1 - E(a, b))^{k_1}$, met k_1 een klein, strikt positief, natuurlijk getal. Anders worden beide opgenomen met een vaste probabiliteit. Ook deze probabiliteit kunnen we gelijk aan 1 stellen.
  - Anders, als H uit meer dan twee objecten bestaat, worden de stimuli voor het opnemen van een object en voor het opnemen van de volledige hoop berekend. De probabiliteiten voor het opnemen van een object en voor het opnemen van de hele hoop, worden gegeven door vergelijkingen (4.1) en (4.2).
- Als de mier reeds een hoop L aan het dragen is
  - Met een vaste probabiliteit wordt de hoop L toegevoegd aan de lijst.
  - Als H uit één enkel object a bestaat en L uit één enkel object b, geldt $A = avg(H) = 1$ en $B = T_W(E(b, a), 1) = E(b, a)$. Aangezien de gevolgen van het ten onrechte samennemen van twee individuele objecten geen zware nadelige gevolgen heeft, kunnen we hier toleranter zijn dan in het geval waarbij $avg(H) = 1$ met $|H| > 1$. In dit geval kiezen we de probabiliteit dat L op H wordt neergelegd gelijk aan $E(b, a)^{k_2}$. Hierbij is k_2 een klein, strikt positief, natuurlijk getal. Als L uit meerdere objecten bestaat, kiezen we ervoor om L niet neer te leggen. Enkel in het onwaarschijnlijke geval dat het centrum van L bijna identiek is aan a en dat de gemiddelde similariteit van L bijna 1 is, zou het evalueren van de vaagregels

leiden tot het neerleggen van L op H. Bovendien zou a in deze situatie bij verdere iteraties toch bijna zeker op L neergelegd worden.

– Anders, als H uit meer dan 1 object bestaat, wordt de stimulus voor het neerleggen bepaald. De probabiliteit dat L op H neergelegd wordt, is dan gegeven door vergelijking (4.3).

Het totale aantal iteraties wordt bepaald in functie van de grootte van de gegevensverzameling X. We komen in paragraaf 4.3 nog terug op de definitie van deze functie. Er wordt dus geen gebruik gemaakt van een globaal stopcriterium. Dit zou immers afbreuk doen aan het lokale karakter van het algoritme. De parameters θ_{obj}, θ_{hoop} en θ_{neer} worden steeds gelijk aan 0.5 gesteld. De andere parameters zullen we verder bepalen.

4.3 Evaluatie

Similariteitsmaat We zullen nu de performantie bespreken van het vaagmieralgoritme voor zowel artificiële gegevensverzamelingen als enkele reële gegevensverzamelingen. Deze laatste zijn vooral belangrijk om de ruisgevoeligheid van het algoritme na te gaan. Steeds zullen we onderstellen dat de te clusteren objecten gekenmerkt worden door een aantal numerieke attributen. We kunnen de objecten m.a.w. beschouwen als vectoren. Het centrum c_H van een hoop $H = \{h_1, h_2 \ldots h_l\}$, met $h_i = (a_{i1}, a_{i2}, \ldots, a_{im})$ voor i in $\{1, 2, \ldots, l\}$, wordt gegeven door $c_H = (\overline{a_1}, \overline{a_2}, \ldots, \overline{a_m})$. Hierbij is $\overline{a_j} = \frac{1}{l}\sum_{i=1}^{l} a_{ij}$, voor j in $\{1, 2, \ldots, m\}$. Om de similariteit tussen twee objecten van de gegevensverzameling X uit te drukken, zullen we gebruik maken van de vaagrelatie E in X, gedefinieerd voor a en b in X als

$$E(a, b) = 1 - \frac{d(a, b)}{d^*(X)}$$

Hierbij is $d(a, b)$ de Euclidische afstand tussen a en b en is $d^*(X)$ de maximale Euclidische afstand tussen objecten uit X. We veronderstellen hierbij dat X ten minste twee verschillende objecten bevat. De reflexiviteit van E volgt uit het feit dat $d(a, a) = 0$, voor alle a in X. Bovendien is ook aan de T_W-transitiviteit voldaan voor deze definitie van E. Voor willekeurige a, b en c in X hebben we immers

$$\begin{aligned}
T_W(E(a,b), E(b,c)) &= \max(0, E(a,b) + E(b,c) - 1) \\
&= \max\left(0, 1 - \frac{d(a,b)}{d^*(X)} + 1 - \frac{d(b,c)}{d^*(X)} - 1\right) \\
&= \max\left(0, 1 - \frac{d(a,b) + d(b,c)}{d^*(X)}\right) \\
&\leq \max\left(0, 1 - \frac{d(a,c)}{d^*(X)}\right) \\
&= 1 - \frac{d(a,c)}{d^*(X)} \\
&= E(a,c)
\end{aligned}$$

We hebben hierbij gebruik gemaakt van het feit dat d voldoet aan de driehoeksongelijkheid en van $\frac{d(a,c)}{d^*(X)} \in [0, 1]$, voor willekeurige a en c in X.

Het bepalen van $d^*(X)$ vereist $|X|(|X|-1)/2$ berekeningen. Een kwadratische uitvoeringstijd zou de toepasbaarheid van het algoritme beperken tot kleine gegevensverzamelingen. We maken daarom gebruik van een benadering van de maximale afstand. Initieel worden $|X|$ paren objecten gekozen waarvoor de afstand berekend wordt. We benaderen $d^*(X)$ dan als het maximum van deze $|X|$ afstanden. Telkens als we, tijdens de uitvoering van het algoritme, een afstand berekenen die groter is dan deze benaderde maximale afstand, wordt de benadering aangepast.

Evaluatiecriteria Evaluatiecriteria voor clusteringsalgoritmen kunnen worden onderverdeeld in twee groepen. Voor de eerste groep is kennis omtrent de correcte classificatie van de objecten niet vereist. Deze criteria vergelijken typisch de similariteit tussen objecten uit dezelfde cluster met de similariteit tussen objecten uit verschillende clusters. De tweede groep evaluatiecriteria vergelijkt de bekomen clusters met de correcte classificatie van de objecten. Het is op deze laatste groep dat we ons hier zullen concentreren. Zij $X = \{x_1, \ldots, x_n\}$ de te clusteren gegevensverzameling. Voor x in X noteren we $k(x)$ voor de klasse waartoe x behoort en $c(x)$ voor de hoop waarin x terecht is gekomen na uitvoering van het algoritme. Zij $L = \{l_1, \ldots, l_m\}$ de lijst met de correcte klassen van objecten, m.a.w. $l_i = \{x | x \in X \text{ en } k(x) = i\}$ voor i in {1,2,... m}. We definiëren de classificatiefout F_c als [59]

$$F_c = \frac{1}{|X|^2} \sum_{1 \leq i,j \leq n} \epsilon_{ij} = \frac{2}{|X|(|X|-1)} \sum_{1 \leq i < j \leq n} \epsilon_{ij}$$

Hierbij is

$$\epsilon_{ij} = \begin{cases} 0 & \text{als } (k(x_i) = k(x_j) \text{ en } c(x_i) = c(x_j)) \text{ of } (k(x_i) \neq k(x_j) \text{ en } c(x_i) \neq c(x_j)) \\ 1 & \text{anders} \end{cases}$$

Initieel zijn alle objecten in een afzonderlijke cluster bevat. In dat geval is steeds voldaan aan $c(x_i) \neq c(x_j)$ voor x_i en x_j in X en $i \neq j$, en dus hebben we $\epsilon_{ij} = 1 \Leftrightarrow k(x_i) = k(x_j)$. De initiële classificatiefout wordt bijgevolg gegeven door

$$F_c = \frac{1}{|X|^2} \sum_{l \in L} \binom{|l|}{2} = \frac{1}{|X|^2} \sum_{l \in L} \frac{|l|(|l|-1)}{2}$$

In het extreme geval dat alle objecten in dezelfde hoop terecht gekomen zijn, hebben we $\epsilon_{ij} = 1 \Leftrightarrow k(x_i) \neq k(x_j)$. De classificatiefout wordt in dit geval gegeven door

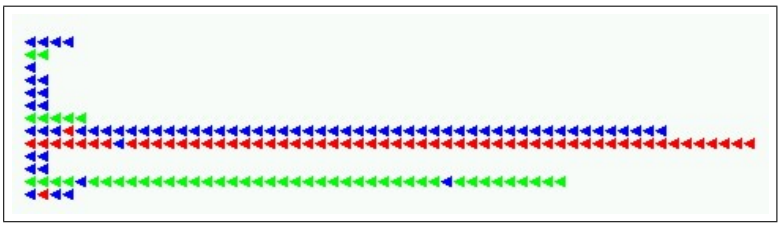
$$F_c = \frac{1}{|X|^2} \sum_{l \in L} \frac{|l|(|X|-|l|)}{2}$$

Voor elk object uit een klasse l zijn er immers $|X|-|l|$ objecten in een andere klasse. Met elke klasse $|l|$ corresponderen er bijgevolg $|l|(|X|-|l|)$ fouten. Wanneer we dit aantal sommeren voor alle hopen, hebben we elke fout twee keer geteld.

Een belangrijk voordeel van dit evaluatiecriterium is dat ook een foutief aantal clusters (zwaar) gepenaliseerd wordt. Aangezien ditzelfde evaluatiecriterium door Monmarché gebruikt werd in [59], zal het ons bovendien toelaten om onze resultaten te vergelijken met de resultaten die in [59] beschreven staan.

Naast de classificatiefout F_c, vormt ook het bekomen aantal clusters een belangrijke indicatie voor de bruikbaarheid van het algoritme. In de meeste gegevensverzamelingen uit de

praktijk komt heel wat ruis voor. Sommige objecten, die we de uitzonderingsgevallen zullen noemen, zijn dan te verschillend van de andere objecten van dezelfde klasse om door een clusteringsalgoritme correct geklasseerd te kunnen worden. Figuur 4.3 illustreert dit fenomeen. In deze figuur komt iedere horizontale rij driehoekjes overeen met een hoop. Hoewel



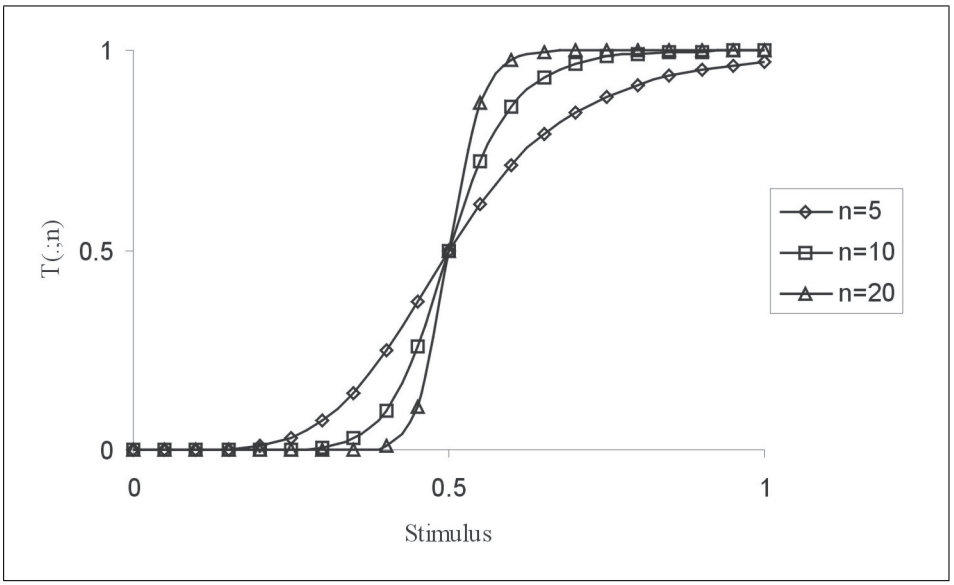
Figuur 4.3: Bepalen van het aantal klassen in de gegevensverzameling

het duidelijk is dat het algoritme de drie klassen in de onderliggende gegevensverzameling ontdekt heeft, is het aantal hopen heel wat groter dan drie. Het expliciet afzonderen van uitzonderingsgevallen is een gewenste eigenschap, aangezien op deze manier de invloed van ruis in de gegevensverzameling op de goede werking van het algoritme kan beperkt worden. Het opnemen van een uitzonderingsgeval in een andere hoop zou immers de berekening van karakteristieken zoals het centrum en de minimale similariteit van die hoop, in sterke mate nadelig beïnvloeden. In concrete implementaties kunnen de uitzonderingsgevallen na afloop van het algoritme aan één van de gevonden clusters worden toegevoegd. Andere mogelijkheden zijn het weglaten van de uitzonderingsgevallen of het samennemen van alle uitzonderingsgevallen in een nieuwe cluster. We kunnen bijvoorbeeld hopen die minder dan 5 objecten bevatten als ruis beschouwen. Voor grote gegevensverzameling kunnen we als alternatief bijvoorbeeld hopen waarvan de grootte minder dan één tiende is van de grootte van de grootste hoop, als ruis beschouwen.

Keuze van de parameters We stelden reeds alle drempelwaarden gelijk aan 0.5. Het is duidelijk dat de parameters n_1, n_2, m_1 en m_2 de hoeveelheid willekeur in de uitvoering van het algoritme beïnvloeden. Wanneer deze parameters grote waarden aannemen, zullen de respectievelijke taken bijna zeker worden uitgevoerd wanneer de stimulus groter is dan 0.5 en bijna zeker niet worden uitgevoerd wanneer de stimulus kleiner is dan 0.5. Wanneer de stimulus gelijk is aan 0.5, zal de kans dat de taak wordt uitgevoerd, wegens de keuze van de drempelwaarden, steeds 0.5 zijn. Dit wordt geïllustreerd in figuur 4.4. Hierbij is $T(.;n)$ de $[0,1]-[0,1]$ afbeelding, gedefinieerd voor s in $[0,1]$ als

$$T(s;n) = \frac{s^n}{s^n + 0.5^n}$$

Voor lage waarden van deze parameters, wordt meer ruimte gelaten voor willekeur. Zoals we in de inleiding van hoofstuk 2 stelden, houdt de hoeveelheid willekeur verband met de kennis van de mier over zijn omgeving. De waarden voor de parameters n_1, n_2, m_1 en m_2 moeten dus de zekerheid weerspiegelen die de gebruikte similariteitsmaat ons geeft omtrent de werkelijke similariteit van de objecten. Deze parameterwaarden zijn bijgevolg afhankelijk van de gebruikte similariteitsmaat. Verder is het duidelijk dat we meer willekeur kunnen tolereren bij het opnemen van objecten of hopen, dan bij het neerleggen van objecten of hopen.



Figuur 4.4: Invloed van de parameters op de hoeveelheid willekeur

Bovendien kunnen we meer willekeur tolereren bij het neerleggen van een individueel object, dan bij het neerleggen van een volledige hoop. We stellen voorop dat $m_1 = m_2 < n_1 < n_2$ moet gelden. Tenzij anders vermeld, werden de experimentele resultaten in dit hoofdstuk bekomen voor $(m_1, m_2, n_1, n_2) = (5, 5, 10, 20)$. De parameters k_1 en k_2 werden gelijk aan 5 gekozen.

4.3.1 Artificiële data

We evalueren nu eerst de werking van het algoritme voor gegevensverzamelingen die we zelf gegenereerd hebben. In het bijzonder zullen we gebruik maken van clusters die gegenereerd werden volgens een normale verdeling. Beschouwen we nu objecten in een m-dimensionale vectorruimte, m.a.w. objecten die bepaald worden door m numerieke attributen. We onderstellen hier steeds dat alle attributen onafhankelijk zijn van elkaar. Zij $\sigma = (\sigma_1, \ldots, \sigma_m)$ en $\mu = (\mu_1, \ldots, \mu_m)$, met $\sigma_i \in [0, +\infty[$ en $\mu_i \in \mathbb{R}$ voor alle i in $\{1, 2, \ldots, m\}$. We noteren dan $N(n; \sigma, \mu)$ voor een cluster van grootte n, waarbij de waarde voor het i^{de} attribuut $(i \in \{1, 2, \ldots, m\})$ van de objecten verdeeld is volgens een normale verdeling met verwachtingswaarde μ_i en standaardafwijking σ_i. Tabel 4.4 geeft een overzicht van de artificiële gegevensverzamelingen die we verder zullen gebruiken. Meestal zullen we de grootte van de clusters n hierbij laten variëren.

Invloed van de parameters De invloed van de parameter n_1 op de classificatiefout F_c wordt geïllustreerd in tabel 4.5. De clustergrootte n werd hierbij gelijk aan 200 gekozen. Deze tabel toont de gemiddelde waarde voor de classificatiefout, vermenigvuldigd met 100, die bekomen werd bij 50 uitvoeringen. Tussen vierkante haken wordt een schatting van de standaardafwijking weergegeven. Deze schatting wordt voor meetwaarden $x_1, x_2, \ldots, x_k$

Benaming	Clusters
Art1	$N(n;(100,100,100),(10,10,10))$, $N(n;(200,200,200),(10,10,10))$
Art2	$N(n;(100,100,100),(20,20,20))$, $N(n;(200,200,200),(20,20,20))$
Art3	$N(n;(100,100,100),(30,30,30))$, $N(n;(200,200,200),(30,30,30))$
Art4	$N(n;(200,100,100),(20,20,20))$, $N(n;(100,200,100),(20,20,20))$, $N(n;(100,100,200),(20,20,20))$
Art5	$N(n;(200,100,100),(20,20,20))$, $N(n;(100,200,100),(20,20,20))$, $N(n;(100,100,200),(20,20,20))$, $N(n;(100,200,200),(20,20,20))$, $N(n;(200,100,200),(20,20,20))$, $N(n;(200,200,100),(20,20,20))$
Art6	$N(n;(100,100,100),(50,50,50))$, $N(n;(200,200,200),(50,50,50))$

Tabel 4.4: Artificiële gegevensverzamelingen

berekend als

$$\sqrt{\frac{\sum_{i=1}^{k}(x_i-\overline{x})^2}{k-1}} = \sqrt{\frac{(\sum_{i=1}^{k}x_i^2)-2\overline{x}(\sum_{i=1}^{k}x_i)+k\overline{x}^2}{k-1}} = \sqrt{\frac{(\sum_{i=1}^{k}x_i^2)-k\overline{x}^2}{k-1}} \tag{4.4}$$

Hierbij is $\overline{x}=\frac{1}{k}\sum_{i=1}^{k}x_i$. De parameters m_1, m_2 en n_2 werden constant gehouden; er geldt dus $(m_1,m_2,n_2)=(5,5,20)$. De belangrijkste conclusie die we kunnen afleiden uit deze resultaten is dat alle waarden uit $[5,20]$ voor de parameter n_1 aanleiding geven tot goede resultaten voor de artificiële gegevensverzamelingen uit tabel 4.4.

	$n_1=5$	$n_1=10$	$n_1=15$	$n_1=20$
Art1	0 [0]	0 [0]	0 [0]	0 [0]
Art2	0 [0]	0.0400 [0.197]	0.140 [0.350]	0.280 [0.572]
Art3	0.220 [0.464]	0.800 [0.857]	1.50 [1.47]	1.92 [1.17]
Art4	0 [0]	0.0400 [0.197]	0.160 [0.421]	0.500 [0.677]
Art5	1.44 [1.52]	0.780 [0.708]	0.940 [0.956]	1.04 [1.29]
Art6	12.9 [5.21]	15.8 [5.43]	16.6 [6.23]	17.9 [5.03]

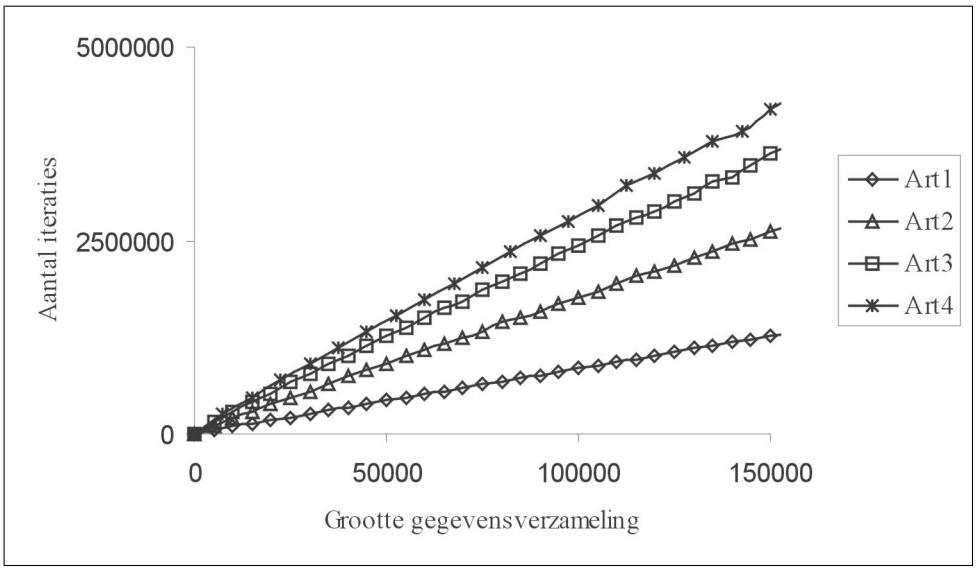
Tabel 4.5: Invloed van de parameter n_1 op de classificatiefout, vermenigvuldigd met 100, na 10^6 iteraties

In tabel 4.6 wordt de invloed van n_2 op de classificatiefout weergegeven. Opnieuw houden we de andere parameterwaarden constant, er geldt m.a.w. $(m_1,m_2,n_1)=(5,5,10)$. We besluiten hieruit dat goede resultaten bekomen worden voor n_2 in $[10,20]$.

	$n_2 = 5$	$n_2 = 10$	$n_2 = 15$	$n_2 = 20$
Art1	0 [0]	0 [0]	0 [0]	0 [0]
Art2	0 [0]	0 [0]	0 [0]	0.0800 [0.340]
Art3	9.14 [19.3]	0.140 [0.350]	0.400 [0.699]	0.700 [0.788]
Art4	18.4 [29.9]	0 [0]	0 [0]	0.0200 [0.141]
Art5	81.7 [8.76]	13.7 [7.49]	2.16 [4.51]	0.680 [0.512]
Art6	41.9 [16.3]	11.2 [2.64]	13.1 [3.77]	15.0 [5.20]

Tabel 4.6: Invloed van de parameter n_2 op de classificatiefout, vermenigvuldigd met 100, na 10^6 iteraties

Aantal iteraties Een belangrijke parameter waarvoor we nog geen waarde gegeven hebben, is het aantal iteraties dat het algoritme moet doorlopen om tot een aanvaardbare oplossing te komen. We tellen hiertoe dit aantal iteraties voor een aantal artificiële gegevensverzamelingen, waarbij we de grootte van deze gegevensverzamelingen laten variëren. In het bijzonder zullen we werken met de gegevensverzamelingen Art1, Art2, Art3 en Art4. Bij deze gegevensverzamelingen is de grootte van alle clusters gelijk aan een parameter n. We beschouwen een oplossing aanvaardbaar wanneer er met elke cluster uit de gegevensverzameling een hoop overeenkomt die ten minste $0.9n$ objecten uit die cluster bevat. Figuur 4.5 toont het resultaat. We leiden hieruit af dat het zinvol is het aantal iteraties voor een gegevensverzameling



Figuur 4.5: Aantal iteraties om tot een aanvaardbare oplossing te komen

X gelijk te stellen aan $c \cdot |X|$ voor een zekere natuurlijke constante c, het aantal iteraties is dus lineair in de grootte van de gegevensverzameling. De waarde van de parameter c is afhankelijk van de toepassing en wordt best voldoende groot gekozen. Het benodigde aantal

iteraties zal immers afhangen van de hoeveelheid ruis in de gegevensverzameling en van het aantal clusters, kenmerken waarover niets gekend is bij aanvang van het algoritme. Voor de eenvoudige gegevensverzamelingen uit figuur 4.5 zou $c = 50$ een goede waarde zijn. Voor reële gegevensverzamelingen met veel ruis, kunnen we verwachten dat bijvoorbeeld $c = 5000$ beter geschikt zal zijn.

Complexiteit Veronderstellen we dus dat het aantal iteraties dat het algoritme moet doorlopen om tot een aanvaardbare oplossing te komen lineair is in de grootte van de gegevensverzameling. Dit betekent echter niet dat ook de uitvoeringstijd lineair is in de grootte van de gegevensverzameling. De uitvoeringstijd van een iteratie is immers afhankelijk van de grootte van de hoop die de mier gekozen heeft. Voor elke hoop zal 1 keer de gemiddelde en minimale similariteit moeten bepaald worden. De tijd die hiervoor nodig is, is lineair in de grootte van de hoop. Veronderstellen we, als een eerste benadering, dat er nooit een individueel object uit een hoop weggenomen wordt en veronderstellen we bovendien dat de gegevensverzameling bestaat uit één enkele cluster van grootte n. We veronderstellen voor de eenvoud dat de grootte van de cluster een macht is van 2. Om een eerste idee te krijgen van de complexiteit, beschouwen we twee mogelijke situaties:

- Initieel bestaan alle hopen uit een individueel object. Wanneer deze objecten nu één voor één worden toegevoegd aan dezelfde hoop, ontstaat achtereenvolgens een hoop van grootte $1, 2, 3, \ldots, n$. De uitvoeringstijd van het algoritme is dan evenredig met

  $$1 + 2 + 3 + \cdots + n = \frac{n(1+n)}{2}$$

  De uitvoeringstijd is dus kwadratisch in de grootte van de gegevensverzameling.

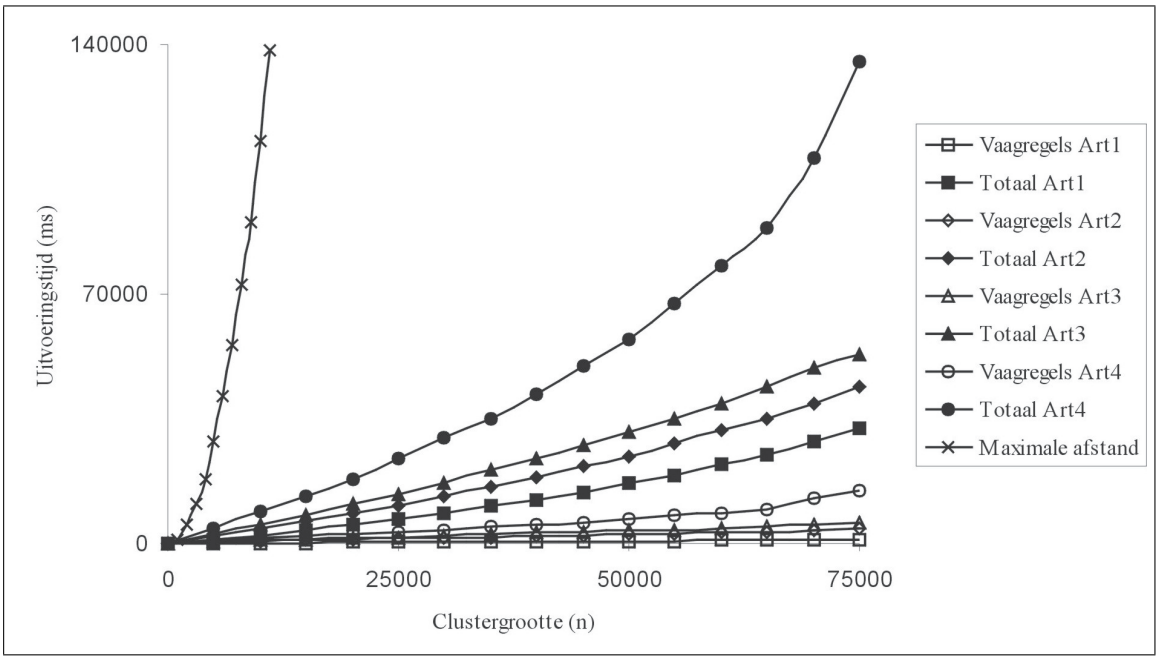
- Veronderstellen we nu dat eerst alle individuele objecten worden samengebracht in $n/2$ hopen van grootte 2. Vervolgens worden deze hopen van grootte 2 samengebracht tot $n/4$ hopen van grootte 4, etc. De uitvoeringstijd van het algoritme is in dit geval evenredig met

  $$n \cdot 1 + \frac{n}{2} \cdot 2 + \frac{n}{4} \cdot 4 + \cdots + \frac{n}{n} \cdot n = n \cdot \log_2 n$$

Figuur 4.6 toont de uitvoeringstijd die nodig was om tot een aanvaardbare oplossing te komen in functie van de clustergrootte n. Zowel de totale uitvoeringstijd als de tijd die gespendeerd werd aan het evalueren van de vaagregels, wordt getoond. Een aanvaardbare oplossing wordt hierbij nog steeds gedefinieerd zoals in de vorige paragraaf. Bovendien toont de grafiek de uitvoeringstijd die nodig is om de exacte maximale afstand tussen objecten uit de gegevensverzameling te bepalen. Hieruit blijkt dat het inderdaad zinvol is om de voorgestelde benadering te hanteren i.p.v. de exacte waarde. Uit deze figuur kunnen we verder afleiden dat de uitvoeringstijd van het algoritme bij benadering evenredig is met $k \cdot \log_2 k$ voor een gegevensverzameling van grootte k. De tijd die gespendeerd wordt aan het evalueren van de vaagregels is lineair in de grootte van de gegevensverzameling. Dit was te verwachten aangezien de uitvoeringstijd voor het evalueren van de vaagregels constant is voor elke iteratie.

4.3.2 Reële data

We zullen gebruik maken van de volgende gegevensverzamelingen van het "UCI Machine Learning Repository" [9], die ook reeds gebruikt werden in o.a. [44] en [59].



Figuur 4.6: Uitvoeringstijd Art2

Wine De "Wine" gegevensverzameling bevat gegevens omtrent een chemische analyse van wijn. De wijn die geanalyseerd werd is afkomstig van drie verschillende wijnboeren uit dezelfde streek in Italië. Een gewenste clustering komt dus overeen met het vinden van drie clusters waarin de objecten uit eenzelfde cluster precies die objecten zijn die overeenkomen met de wijn die afkomstig is van eenzelfde wijnboer. Een object wordt beschreven a.d.h.v. 13 numerieke attributen en komt overeen met de meetwaarden bij een bepaald staal. Het aantal objecten in elke klasse (m.a.w. het aantal stalen van elke wijnboer) wordt gegeven door 59, 71 en 48.

Iris De "Iris" gegevensverzameling bevat gegevens omtrent drie soorten Iris-planten. Elk object wordt beschreven door 4 numerieke attributen en er komen precies 50 objecten voor uit elke klasse.

Glass De "Glass" gegevensverzameling bevat gegevens omtrent verschillende glassoorten. In de gegevensverzameling komen 2 hoofdklassen voor: glas dat gebruikt wordt voor ruiten (Eng. window glass) en ander glas. De eerste klasse kan worden onderverdeeld in 4 subklassen, waarvan er 3 effectief voorkomen in de gegevensverzameling. De tweede klasse kan worden onderverdeeld in 3 subklassen die alle voorkomen in de gegevensverzameling. Zowel het vinden van twee clusters als het vinden van 6 clusters kan dus beschouwd worden als een goed resultaat. Bij de verdere bespreking zullen we de classificatiefout berekenen m.b.t. de twee hoofdklassen.

Invloed van de parameters

We bespreken eerst de invloed van de parameter n_1 op de classificatiefout F_c. De gemiddelde waarde voor F_c, vermenigvuldigd met 100, die bekomen werd na 10^6 iteraties

bij 50 uitvoeringen, en de standaardafwijking worden getoond in tabel 4.7. Hierbij werd $(m_1, m_2, n_2) = (5, 5, 20)$ constant gehouden. Hieruit blijkt dat $n_1 = 10$ inderdaad een goede keuze is. Bovendien is duidelijk dat kleine verschillen in de waarde van n_1 geen grote invloed hebben op het resultaat. Zelfs voor $n_1 = 20$ worden behoorlijke resultaten behaald. In [59] werden voor de "Wine", "Iris" en "Glass" gegevensverzamelingen respectievelijk de waarden 0.51, 0.19 en 0.40 gevonden voor de classificatiefout F_c. De classificatiefout voor de "Glass" gegevensverzameling werd in [59] echter berekend m.b.t. de 6 subklassen. Het vaagmieralgoritme vindt steeds de twee hoofdklassen, het algoritme van Monmarché slaagt er niet in om de 6 subklassen of de 2 hoofdklassen te identificeren op een betrouwbare manier. Ook voor de "Wine" gegevensverzameling is het resultaat van het vaagmieralgoritme stukken beter. Voor de "Iris" gegevensverzameling vinden we een kleine verbetering.

	$n_1 = 5$	$n_1 = 10$	$n_1 = 15$	$n_1 = 20$
Wine	50.5 [16.9]	13.3 [2.78]	14.4 [1.90]	16.0 [1.89]
Iris	17.9 [3.70]	16.0 [2.65]	17.7 [2.22]	17.7 [2.51]
Glass	16.9 [6.51]	12.7 [2.03]	13.4 [1.51]	14.6 [1.38]

Tabel 4.7: Invloed van de parameter n_1 op 100 F_c na 10^6 iteraties

Tabel 4.8 toont de invloed van de parameter n_2. We zien hierbij dat voor n_2 in $[15, 20]$ goede resultaten bekomen worden. Zelfs voor $n_2 = 10$ worden nog behoorlijke resultaten gevonden. Ten slotte wordt in 4.9 de invloed van de parameters m_1 en m_2 geïllustreerd. Hieruit leiden we af dat deze parameterwaarden geen grote invloed hebben op het resultaat wanneer deze voldoende klein gekozen worden, bijvoorbeeld in het interval $[5, 10]$.

	$n_2 = 5$	$n_2 = 10$	$n_2 = 15$	$n_2 = 20$
Wine	65.0 [0.246]	21.7 [21.9]	11.4 [2.39]	12.8 [2.39]
Iris	24.6 [10.5]	22.0 [0.494]	20.5 [1.29]	16.3 [2.41]
Glass	34.5 [1.84]	11.7 [2.82]	11.8 [1.15]	12.2 [1.72]

Tabel 4.8: Invloed van de parameter n_2 op 100 F_c na 10^6 iteraties

Aantal klassen

In tabel 4.10 wordt het aantal gevonden klassen bij 50 uitvoeringen van het algoritme samengevat. We hebben hierbij hopen die uit minder dan 5 objecten bestaan als ruis beschouwd. We hebben m.a.w. het aantal hopen geteld met grootte ten minste 5. We zien dat bij de drie gegevensverzamelingen het meest aantal keer het correcte aantal klassen gevonden wordt. Bij de "Iris" gegevensverzameling zien we dat één keer te weinig klassen gevonden worden. De twee klassen met overlap werden in dit geval samengenomen. Bij heel wat uitvoeringen wordt een te groot aantal klassen gevonden, wat verklaard kan worden door de aanwezigheid van

	$(m_1, m_2) = (5, 5)$	$(m_1, m_2) = (10, 10)$	$(m_1, m_2) = (15, 15)$	$(m_1, m_2) = (20, 20)$
Wine	13.3 [1.78]	12.4 [2.13]	13.2 [2.25]	13.1 [1.94]
Iris	17.3 [3.12]	16.22 [2.57]	16.7 [2.33]	16.7 [2.87]
Glass	12.6 [1.68]	12.5 [1.18]	12.7 [1.62]	12.3 [1.76]

Tabel 4.9: Invloed van de parameters m_1 en m_2 op 100 F_c na 10^6 iteraties

ruis in de gegevensverzameling. Door deze aanwezigheid van ruis, ontstaan kleine hopen van uitzonderingsgevallen, zoals geïllustreerd wordt in figuur 4.3. Soms bevatten deze hopen meer dan vier objecten, waardoor ze niet als ruis herkend worden.

	2 klassen	3 klassen	4 klassen	5 klassen
Wine	0	32	15	3
Iris	1	23	21	5
Glass	30	11	7	2

Tabel 4.10: Aantal gevonden klassen bij 50 uitvoeringen

Hoofdstuk 5

Vergelijken van documenten en termen

5.1 Inleiding

We bespreken in dit hoofdstuk in eerste instantie hoe we similariteit tussen documenten kunnen uitdrukken. Hierbij wordt er steeds een abstractie gemaakt van een document. De meest gebruikte aanpak bestaat erin geen rekening te houden met de volgorde van de termen in een document en enkel het aantal voorkomens van elke term in een document te beschouwen. Een document wordt dan voorgesteld als een multiverzameling, m.a.w. een $\mathcal{T} - \mathbb{N}$ afbeelding, met $\mathcal{T}$ het universum van alle mogelijke termen. In het Engels spreekt men van de "Bag of words-approach". Zoals we zullen zien, is deze multiverzameling in de praktijk enkel een conceptuele tussenstap om het document bijvoorbeeld als een vector of als een vaagverzameling voor te stellen. We zullen ook bespreken hoe we m.b.v. inclusiematen kunnen nagaan in welke mate een document specifieker is dan een ander document.

Anderzijds zullen we ook nagaan hoe we similariteit en de relatie "specifieker dan" voor termen kunnen modelleren. Deze relaties tussen termen zullen ons in staat stellen om documenten op een rijkere manier te vergelijken. Op deze manier kunnen twee documenten die geen enkele term gemeenschappelijk hebben, toch een van nul verschillende similariteit krijgen. Bij de ingevoerde similariteits- en inclusiematen zullen we tevens nagaan of de reflexiviteit en de T_W-transitiviteit die we ondersteld hebben bij het opstellen van het vaagmieralgoritme vervuld zijn.

5.2 Het vectorruimtemodel

5.2.1 Documenten als vectoren

Zij $\mathcal{D} = \{d_1, d_2, \ldots, d_n\}$ een eindige, niet ledige verzameling documenten. We noteren de (eindige) verzameling van alle termen die voorkomen in de documenten uit $\mathcal{D}$ als $\mathcal{T} = \{t_1, t_2, \ldots, t_m\}$. Zij $A = (a_{ij})$ een $|\mathcal{T}| \times |\mathcal{D}|$ matrix waarbij het element a_{ij} op de i^{de} rij en in de j^{de} kolom ($i \in \{1, 2, \ldots, m\}$, $j \in \{1, 2, \ldots, n\}$) het belang aangeeft van de term t_i in het document d_j; a_{ij} wordt het gewicht van t_i in d_j genoemd. We onderstellen hierbij $a_{ij} \in [0, +\infty[$ voor alle i in $\{1, 2, \ldots, m\}$ en alle j in $\{1, 2 \ldots, n\}$. De matrix A wordt de term-documentmatrix genoemd. De rijvectoren van de matrix A stellen termen voor, de ko-

lomvectoren documenten. Verschillende mogelijkheden bestaan voor het toekennen van de gewichten:

- In het binair model krijgt elke term die voorkomt in een document gewicht 1. Alle andere termen krijgen gewicht 0. Dit eenvoudige model laat niet toe om een onderscheid te maken tussen belangrijke en minder belangrijke termen.

- In het TF-model (Eng. term frequency) is het gewicht van elke term in een document gelijk aan het aantal keer deze term voorkomt in dit document. Dit model steunt op de veronderstelling dat de frequentie van voorkomen van een term in een document, een goede indicatie is voor het belang ervan. Deze veronderstelling wordt soms in twijfel getrokken. Wanneer een document bijvoorbeeld over één enkel onderwerp gaat, is het niet noodzakelijk om dit onderwerp telkens te herhalen in de tekst [50]. Niettemin lijkt deze veronderstelling in het algemeen wel op te gaan.

- Het $TF{-}IDF$-model (Eng. term frequency - inverse document frequency) is een aanpassing van het TF-model waarbij ook het aantal documenten uit $\mathcal{D}$ waarin een bepaalde term voorkomt in rekening wordt gebracht. Noem df_i het aantal documenten waarin de term t_i voorkomt en noem tf_{ij} het aantal keer dat t_i voorkomt in document d_j. Het gewicht van t_i in d_j wordt dan bijvoorbeeld gegeven door [50]

$$a_{ij} = \begin{cases} (1 + tf_{ij}) \log_2(\frac{n}{df_i}) & \text{als } tf_{ij} > 0 \\ 0 & \text{anders} \end{cases}$$

  Er bestaan verschillende varianten hierop. Het uitgangspunt is steeds dat termen die in weinig documenten voorkomen belangrijker zijn dan termen die in veel documenten voorkomen.

Ook de plaats waar een term voorkomt in een document wordt soms in rekening gebracht bij de bepaling van de gewichten. Termen die in een titel voorkomen krijgen dan bijvoorbeeld een hoger gewicht.

Aangezien we documenten als vectoren representeren zijn alle similariteits- en afstandsmaten voor vectoren mogelijke kandidaten om de similariteit tussen documenten te modelleren. Het is echter algemeen gekend dat de Euclidische afstand geen goede keuze is. Dit wordt o.m. bevestigd door experimentele resultaten in [71]. De reden hiervoor is dat het niet voorkomen van een term in twee documenten dezelfde impact heeft op de dissimilariteit tussen die documenten als het samen voorkomen van een term in die documenten. Veruit de meest gebruikte similariteitsmaat voor documenten is de cosinussimilariteit sim_{cos} die voor de documenten d_i en d_j $(1 \leq i, j \leq n\})$ gedefinieerd wordt als

$$sim_{cos}(d_i, d_j) = \frac{\sum_{l=1}^{m} a_{li} \cdot a_{lj}}{\left(\sqrt{\sum_{l=1}^{m} a_{li}^2}\right)\left(\sqrt{\sum_{l=1}^{m} a_{lj}^2}\right)}$$

De cosinussimilariteit is m.a.w. de cosinus van de hoek tussen de vector die correspondeert met het document d_i en de vector die correspondeert met het document d_j. Hierbij veronderstellen we dat geen enkele rijvector van de matrix A de nulvector is, m.a.w. dat elk document minstens

één term bevat. Merk op dat de meeste componenten van de documentvectoren 0 zullen zijn. Een efficiënte implementatie zal dit uitbuiten, waardoor de uitvoeringstijd die nodig is om de cosinussimilariteit te bepalen tussen twee documenten, evenredig is met de lengte van deze twee documenten i.p.v. evenredig met de grootte van de termenverzameling.

Eigenschap 18 (Cauchy-Schwartz ongelijkheid). *Zij a en b twee n-dimensionale vectoren met reële componenten, met n een strikt positief natuurlijk getal. Dan geldt er*

$$\left(\sum_{i=1}^{n} a_i b_i\right)^2 \leq \left(\sum_{i=1}^{n} a_i^2\right)\left(\sum_{i=1}^{n} b_i^2\right)$$

De gelijkheid geldt a.s.a. er een reële constante λ bestaat waarvoor $a = \lambda b$

Voor het bewijs verwijzen we bijvoorbeeld naar [66]. Er volgt hieruit onmiddellijk dat voor willekeurige documenten d_i en d_j $(1 \leq i, j \leq n\})$ voldaan is aan $sim_{cos}(d_i, d_j) \leq 1$. Bovendien volgt uit de definitie van sim_{cos} en het feit dat alle gewichten niet-negatief zijn, dat steeds voldaan is aan $0 \leq sim_{cos}(d_i, d_j)$. Bijgevolg is sim_{cos} een reflexieve, symmetrische vaagrelatie in het universum van de documenten. Deze vaagrelatie is evenwel niet T_W-transitief. We beschouwen hiertoe $\mathcal{D} = \{d_1, d_2, d_3\}$, $\mathcal{T} = \{t_1, t_2\}$, de matrix A wordt gegeven door

$$A = \begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \end{pmatrix}$$

dan hebben we

$$\begin{aligned} T_W(sim_{cos}(d_1, d_2), sim_{cos}(d_2, d_3)) &= sim_{cos}(d_1, d_2) + sim_{cos}(d_2, d_3) - 1 \\ &= \frac{1+0}{1 \cdot \sqrt{2}} + \frac{0+1}{\sqrt{2} \cdot 1} - 1 \\ &= \sqrt{2} - 1 \\ &> 0 \\ &= sim_{cos}(d_1, d_3) \end{aligned}$$

5.2.2 Dimensiereductie

Het vectorruimte model uit de vorige paragraaf houdt geen rekening met het bestaan van synoniemen. Enkel het letterlijk voorkomen van dezelfde termen in twee documenten heeft een invloed op de similariteit. Alle termen worden, wanneer we gebruik maken van de cosinussimilariteit, onafhankelijk verondersteld. Wanneer we een documentenverzameling clusteren m.b.v. een dergelijke similariteitsmaat, ontstaat een groot aantal, kleine clusters. De kolommen van de matrix A spannen een vectorruimte op, die we de documentruimte zullen noemen. Een veelgebruikte oplossing om afhankelijkheden tussen termen (impliciet) in rekening te brengen, bestaat erin de dimensie van de documentruimte drastisch te verlagen. De componenten van de documentvectoren in deze gereduceerde documentruime stellen dan concepten voor, eerder dan termen. Bovendien kan de cosinussimilariteit sneller berekend worden in deze gereduceerde documentruimte. Dit voordeel is vooral belangrijk voor grote documenten en bij toepassingen waarbij de dimensiereductie vooraf kan berekend worden zoals bij zoekmachines. Technieken om deze dimensiereductie aan te passen wanneer de documentenverzameling gewijzigd wordt, werden hiertoe ontwikkeld [7]. We bespreken nu een tweetal mogelijkheden om deze dimensiereductie te realiseren.

Latente semantische indexering (LSI)

(Eng. Latent Semantic Indexing) De LSI-methode realiseert een dimensiereductie van de documentenruimte door gebruik te maken van een zogenaamde singulaire-waardedecompositie van de term-documentmatrix A. Deze methode is afhankelijk van een strikt positieve, natuurlijke constante k die de dimensie van de resulterende documentruimte voorstelt.

Definitie 75 (Orthogonaal). *[79] Een n-dimensionale vierkante matrix A wordt orthogonaal genoemd a.s.a voldaan is aan $AA^T = A^TA = I_n$, m.a.w. wanneer geldt dat $A^T = A^{-1}$. Hierbij is I_n de eenheidsmatrix van dimensie n.*

Eigenschap 19 (Singulaire-waardedecompositie). *[79] Zij A een $m \times k$ matrix. Dan bestaat er een orthogonale $m \times m$ matrix U, een orthogonale $k \times k$ matrix V en een $m \times k$ matrix Σ met het element op de i^{de} rij en de i^{de} kolom gelijk aan $\lambda_i \geq 0$ ($i \in \{1, 2, \ldots, \min(m, k)\}$) en alle andere elementen 0, waarvoor voldaan is aan*

$$A = U\Sigma V^T$$

De constanten $\lambda_1, \ldots, \lambda_{\min(k,m)}$ worden de singulaire waarden van A genoemd, de kolomvectoren van U worden de links-singulaire vectoren genoemd en de kolomvectoren van V de rechts-singulaire vectoren. Noem u_i de i^{de} kolomvector van U en v_i de i^{de} kolomvector van V ($i \in \{1, 2, \ldots, \min(k, m)\}$). Het drietal (u_i, v_i, λ_i) wordt dan een singulair triplet genoemd ([55]), u_i wordt de links-singulaire vector genoemd corresponderend met λ_i, v_i de rechts-singulaire vector.

Zij $A = U\Sigma V^T$ nu de singulaire-waardedecompositie van de term-documentmatrix A waarbij voor de diagonaalelementen $\lambda_1, \lambda_2, \ldots \lambda_{\min(m,n)}$ van Σ voldaan is aan $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_{\min(m,n)}$. De matrix Σ kan, door gebruik te maken van rij- en kolompermutaties, steeds zo gekozen worden dat hieraan voldaan is. Zij U_k nu de $m \times k$ matrix die bestaat uit de eerste k kolomvectoren van U, Σ_k een $k \times k$ diagonale matrix, waarbij de diagonaalelementen de k grootste singulaire waarden zijn in dalende volgorde en zij V_k de $n \times k$ matrix die bestaat uit de eerste k kolomvectoren van V. Dan is $A_k = U_k\Sigma_k V_k^T$ een benadering van A waarbij de dimensie van de vectorruimte die opgespannen wordt door de kolommen gereduceerd is tot k. Er geldt dat [7]

$$\min\{\|A - B\|_F \mid B \text{ is een } m \times n \text{ matrix van rang } k\} = \|A - A_k\|_F$$

Hierbij is $\|.\|_F$ de Frobenius norm, gedefinieerd voor een $m \times n$ matrix $B = (b_{ij})$ als

$$\|B\|_F = \sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n} b_{ij}^2}$$

De rijvectoren van V en van V_k stellen documenten voor, waarvan de componenten concepten voorstellen. Het uitgangspunt van LSI is dat deze concepten de latente (m.a.w. verborgen) structuur van de termenverzameling weerspiegelen in die zin dat verwante termen door hetzelfde concept worden voorgesteld. Hierbij worden de concepten die corresponderen met kleine singulaire waarden als ruis beschouwd. Op dezelfde manier stellen de rijvectoren van U en U_k termen voor. Door documenten (resp. termen) te clusteren a.d.h.v. de vectorvoorstelling die bepaald wordt door V_k (resp. U_k) i.p.v. de vectorvoorstelling bepaald door de

term-documentmatrix A wordt dus rekening gehouden met de relaties die bestaan tussen de verschillende termen waarbij de voorstelling bovendien minder ruis bevat.

De bepaling van de singulaire tripletten van een matrix A is kubisch in het aantal tripletten dat berekend moet worden [55]. De bepaling van een individueel singulair triplet is echter enkel afhankelijk van de voorgaande tripletten (m.a.w. de tripletten die corresponderen met grotere singulaire waarden), zodat de bepaling van een klein aantal singulaire tripletten toch redelijk efficiënt kan gebeuren [55]. Het is dus, omwille van efficiëntieredenen, belangrijk om de dimensie k niet te hoog te kiezen. Wanneer k te laag gekozen wordt, blijft onvoldoende informatie over om documenten betrouwbaar te kunnen clusteren. Uit experimentele resultaten in [55] blijkt dat zowel wanneer k te hoog als wanneer k te laag gekozen wordt, de kwaliteit van de bekomen clusters drastisch vermindert. Voor optimale waarden van k wordt wel een significante verbetering vastgesteld t.o.v. het originele vectorruimtemodel.

Covariantiematrix-analyse

Deze methode is gebaseerd op een analyse van de hoofdcomponenten (Eng. principal component analysis (PCA)) van de covariantiematrix van de documentenverzameling . We vermelden eerst de nodige definities en eigenschappen.

Definitie 76 (Eigenwaarde). *[79] Een n-dimensionale vierkante matrix A heeft een eigenwaarde $\lambda \in \mathbb{R}$, met corresponderende n-dimensionale eigenvector $x \neq 0$, wanneer voldaan is aan $Ax = \lambda x$.*

Eigenschap 20. *[79] Een symmetrische n-dimensionale vierkante matrix A heeft n paren van eigenwaarden en eigenvectoren. De eigenvectoren kunnen genormaliseerd gekozen worden. De genormaliseerde eigenvectoren zijn uniek wanneer alle eigenwaarden verschillend zijn.*

Eigenschap 21 (Spectraaldecompositie). *[79] De spectraaldecompositie van een symmetrische n-dimensionale vierkante matrix A, wordt gegeven door*

$$A = \lambda_1 e_1 e_1^T + \cdots + \lambda_n e_n e_n^T = V \Sigma V^T$$

met $(\lambda_1, e_1), \ldots, (\lambda_n, e_n)$ de paren van eigenwaarden en genormaliseerde eigenvectoren van A. Hierbij is Σ de n-dimensionale diagonaalmatrix met λ_i als i^{de} diagonaalelement en V de n-dimensionale vierkante matrix met e_i als i^{de} kolom ($i \in \{1, 2, \ldots, n\}$).

Definitie 77 (Toevalsvector). *[79] Zij $X_1, X_2, \ldots, X_n$ toevalsveranderlijken, dan wordt $X = (X_1, X_2, \ldots, X_n)^T$ een toevalsvector genoemd.*

Definitie 78 (Covariantiematrix). *[79] Zij $X = (X_1, X_2, \ldots, X_n)^T$ een toevalsvector en noteren we $E[A]$ voor de verwachtingswaarde van een toevalsveranderlijke A. De covariantiematrix van een toevalsvector X is dan de $n \times n$ matrix Σ waarbij het element op de i^{de} rij en in de j^{de} kolom gegeven wordt door $E[(X_i - E[X_i])(X_j - E[X_j])]$.*

Definitie 79 (Hoofdcomponent). *[79] Zij $X = (X_1, X_2, \ldots, X_n)^T$ een toevalsvector met covariantiematrix Σ. De eerste hoofdcomponent (Eng. principal component) van X is dan de lineaire combinatie*

$$l_1^T X = l_{11} X_1 + l_{21} X_2 + \cdots + l_{n1} X_n$$

die $Var(l_1^T X) = l_1^T \Sigma l_1$ *maximaliseert. Meer algemeen definiëren we de* h^{de} *hoofdcomponent van* X *(*$h \in \{1, 2, \ldots, n\}$*) als de lineaire combinatie*

$$l_h^T X = l_{1h} X_1 + l_{1h} X_2 + \cdots + l_{nh} X_n$$

die $Var(l_h^T X) = l_h^T \Sigma l_h$ *maximaliseert en waarvoor* $Cov(l_i^T X, l_h^T X) = l_i^T \Sigma l_h = 0$ *voor alle* i *in* $\{1, 2, \ldots h-1\}$*.*

Eigenschap 22. *[79] Zij* Σ *de covariantiematrix van een toevalsvector* $X = (X_1, \ldots, X_n)^T$ *en zij* $(\lambda_1, e_1), (\lambda_2, e_2), \ldots (\lambda_n, e_n)$ *de eigenwaarde-eigenvector paren van* Σ *waarbij* $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n \geq 0$ *en waarbij* $\{e_1, e_2, \ldots e_n\}$ *een orthonormale verzameling vectoren is, m.a.w.* $e_i e_i^T = 1$ *voor alle* i *in* $\{1, 2, \ldots, n\}$ *en* $e_i e_j^T = 0$ *voor alle* i *en* j *in* $\{1, 2, \ldots n\}$ *waarvoor* $i \neq j$*. Noteer de coördinaten van* e_i *als* $e_i = (e_{1i}, e_{2i} \ldots, e_{ni})^T$ *(*$i \in \{1, 2, \ldots n\}$*). De* h^{de} *hoofdcomponent van* X *wordt dan gegeven door (*$h \in \{1, 2, \ldots n\}$*)*

$$Y_h = e_h^T X = e_{1h} X_1 + e_{2h} X_2 + \cdots + e_{nh} X_n$$

We kunnen nu een document beschouwen als een toevalsvector $X = (X_1, X_2, \ldots, X_m)^T$ waarbij X_i ($i \in \{1, 2, \ldots, m\}$) de toevalsveranderlijke is die correspondeert met het gewicht van de term t_i in een document. De corresponderende covariantiematrix is de $m \times m$ matrix Σ waarbij het element σ_{ij} op de i^{de} rij en de j^{de} kolom ($i, j \in \{1, 2, \ldots, m\}$) gegeven wordt door [50]

$$\frac{1}{n} \sum_{l=1}^{n} (a_{il} \cdot a_{jl} - \overline{a_i} \cdot \overline{a_j})$$

Hierbij is $A = (a_{ij})$ nog steeds de $m \times n$ term-documentmatrix en definiëren we $\overline{a_i} = \frac{1}{n} \sum_{l=1}^{n} a_{il}$. Zij $(\lambda_1, e_1), (\lambda_2, e_2), \ldots, (\lambda_m, e_m)$ de eigenwaarde-eigenvector paren van Σ waarbij $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n \geq 0$ en waarbij $\{e_1, e_2, \ldots e_n\}$ een orthonormale verzameling vectoren is en zij V de $m \times m$ matrix met e_i als i^{de} kolom ($i \in \{1, 2, \ldots, m\}$). Wegens eigenschap 22 is $B = V^T A$ de $m \times n$ matrix waarbij het element op de i^{de} rij en de j^{de} kolom de waarde voor de i^{de} hoofdcomponent van het document d_j voorstelt. Opnieuw worden de hoofdcomponenten die corresponderen met kleine eigenwaarden als ruis beschouwd. Zij k opnieuw een strikt positieve, natuurlijke constante die de dimensie van de resulterende documentruimte voorstelt. Zij V_k de $m \times k$ matrix die bestaat uit de eerste k kolommen van V en zij $B_k = V_k^T A$. We kunnen de kolomvectoren van B_k beschouwen als documentvectoren waarbij de componenten de belangrijkste concepten uit de documentverzameling voorstellen i.p.v. termen. Documenten kunnen nu geclusterd worden door de cosinussimilariteit te berekenen op basis van de matrix B_k i.p.v. op basis van de originele term-documentmatrix A.

De werking van LSI en covariantiematrix-analyse op deze manier is heel analoog. Het is echter ook mogelijk om de hoofdcomponenten rechtstreeks, zonder het gebruik van een similariteits- of afstandsmaat, te gebruiken om de documentverzameling te clusteren. Een eerste mogelijkheid maakt enkel gebruik van de eerste hoofdcomponent [24]. De $1 \times n$ matrix $B_1 = (b_i)$ beschrijft de documenten a.d.h.v. één enkele component. Op deze manier worden documenten beschreven als een punt van een rechte. De documentverzameling $\mathcal{D}$ kan dan gesplitst worden in de twee deelverzamelingen $\mathcal{D}_1$ en $\mathcal{D}_2$, gedefinieerd door

$$\begin{aligned} \mathcal{D}_1 &= \{d_i | 1 \leq i \leq n \text{ en } b_i \leq \mu\} \\ \mathcal{D}_2 &= \{d_i | 1 \leq i \leq n \text{ en } b_i > \mu\} \end{aligned}$$

waarbij bijvoorbeeld $\mu = \frac{1}{n}\sum_{i=1}^{n} b_i$. De verzamelingen $\mathcal{D}_1$ en $\mathcal{D}_2$ kunnen dan analoog verder worden opgesplitst tot een bepaald stopcriterium bereikt is. In [24] wordt een gelijkaardig algoritme voorgesteld waarbij gebruik gemaakt wordt van de eerste twee hoofdcomponenten. Er wordt een algoritme opgesteld om documenten die beschreven worden als punten in een vlak op te splitsen in twee groepen.

5.2.3 Conclusies

Het representeren van documenten door vectoren vertaalt het probleem van het bepalen van de similariteit tussen documenten naar een probleem uit de lineaire algebra. Wanneer gebruik gemaakt wordt van de cosinussimilariteit en de originele term-documentmatrix, worden alle termen onafhankelijk verondersteld. Door de dimensie van de documentruimte te reduceren wordt deze veronderstelling opgeheven en worden afhankelijkheden tussen termen impliciet in rekening gebracht. Hoewel heel wat experimentele resultaten bevestigen dat deze methoden het klassieke vectorruimte model voor optimale dimensies overtreffen in kwaliteit, blijken theoretische verklaringen moeilijk te vinden. In [63] wordt d.m.v. een probabilistisch model aangetoond dat LSI onder bepaalde voorwaarden de semantische structuur van de documentverzameling weerspiegelt. Het is echter onduidelijk op welke manier dit gebeurt. De keuze van de dimensie van de gereduceerde documentruimte heeft een grote invloed op het resultaat. Nochtans blijkt het instellen van deze parameter een moeilijke opdracht. Bovendien lijken deze dimensie-reductiemethoden niet onmiddellijk bruikbaar om de relatie "specifieker dan" voor termen te benutten en is het niet mogelijk om gebruik te maken van onafhankelijk gecreëerde synoniemenwoordenboeken. Het feit dat de cosinussimilariteit niet T_W-transitief is, vormt een bijkomende beperking voor de toepassing van het vaagmieralgoritme. Ten slotte vermelden we nog de hoge computationele kost die verbonden is met de dimensiereductie, waardoor de methode niet bruikbaar is voor grote documentverzamelingen bij toepassingen waarbij de gebruiker moet wachten op het resultaat van deze berekening, zoals het clusteren van zoekresultaten.

5.3 Documenten als vaagverzamelingen

Zij $\mathcal{D} = \{d_1, d_2, \ldots, d_n\}$ nog steeds een eindige, niet-ledige documentverzameling en $\mathcal{T} = \{t_1, t_2, \ldots, t_m\}$ de eindige verzameling van termen die voorkomen in de documenten uit $\mathcal{D}$. Zij R nu een vaagrelatie van $\mathcal{D}$ naar $\mathcal{T}$, m.a.w. $R \in \mathcal{F}(\mathcal{D} \times \mathcal{T})$, zodat voor d in $\mathcal{D}$ en t in $\mathcal{T}$, $R(d,t)$ de mate aangeeft waarin de term t relevant is voor het document d. De waarde voor $R(d,t)$ kan bepaald worden op een analoge manier als de gewichten in het vectorruimte model, waarbij we er hier natuurlijk wel moeten voor zorgen dat steeds voldaan is aan $R(d,t) \in [0,1]$. Voor elke d in $\mathcal{D}$ stelt dR een vaagverzameling in $\mathcal{T}$ voor, die we kunnen gebruiken als representatie voor het document. Analoog stelt Rt voor elke t in $\mathcal{T}$ een vaagverzameling in $\mathcal{D}$ voor die we kunnen gebruiken als represenatie voor de term. De similariteit tussen documenten en de relatie "specifieker dan", kunnen gemodelleerd worden a.d.h.v. vaagrelaties in $\mathcal{D}$. Om deze vaagrelaties op te stellen zal gebruik gemaakt worden van vaagrelaties in $\mathcal{F}(\mathcal{T})$ die gebruik maken van de voorgestelde representatie van een document. We onderstellen verder steeds dat elk document ten minste één term bevat, m.a.w. $\sum_{t \in \mathcal{T}} R(d,t) > 0$, voor elke d in $\mathcal{D}$.

5.3.1 Similariteit

Het begrip tolerantierelatie dat we in hoofdstuk 3 gedefinieerd hebben, kan worden veralgemeend voor vaagrelaties. We beschouwen verder de volgende definitie.

Definitie 80 (Tolerantierelatie). *Een vaagrelatie $R \in \mathcal{F}(X^2)$ in een universum X wordt een tolerantierelatie genoemd als voldaan is aan*

reflexiviteit $(\forall x \in X)(R(x,x) = 1)$

symmetrie $(\forall (x,y) \in X^2)(R(x,y) = R(y,x))$

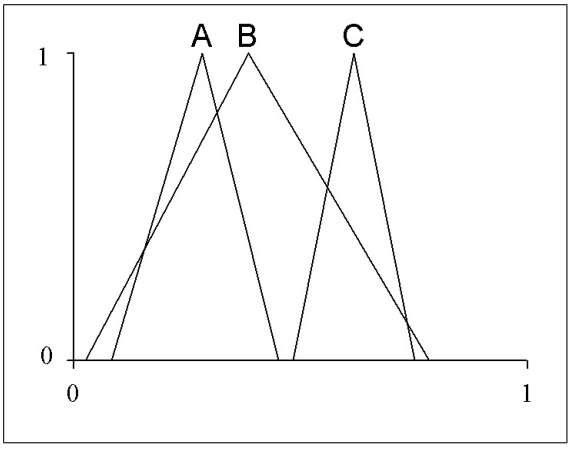
Een relatie die de similariteit tussen objecten in een zeker universum moet uitdrukken, zal in elk geval een tolerantierelatie moeten zijn. Het is echter gebruikelijk om bijkomende voorwaarden op te leggen, aangezien lang niet elke tolerantierelatie geschikt is om similariteit uit te drukken. Het begrip T-equivalentierelatie, dat we in hoofdstuk 3 hebben gedefinieerd, is een speciaal geval van een tolerantierelatie, waarbij ook T-transitiviteit geëist wordt; hierbij is T een triangulaire norm. Definiëren we de $[0,1]^2 - [0,1]$ afbeelding ε_T nu voor a en b in $[0,1]$ als

$$\varepsilon_T(a,b) = \min(I_T(a,b), I_T(b,a)) \tag{5.1}$$

Hierbij is I_T de residuele implicator die correspondeert met T. Wegens (3.3) geldt voor elke a en b in $[0,1]$ dat $I_T(a,b) = 1$ of $I_T(b,a) = 1$. Bijgevolg kunnen we het minimum in (5.1) vervangen door een willekeurige t-norm. De vaagrelatie E_T in $\mathcal{F}(X)$, met X een universum, die voor elementen A en B van $\mathcal{F}(X)$ gedefinieerd wordt door

$$E_T(A,B) = \inf_{x \in X} \varepsilon_T(A(x), B(x)) \tag{5.2}$$

is een T-equivalentie voor elke links-continue t-norm T [77]. Figuur 5.1 illustreert dat een T-equivalentie soms ongewenste eigenschappen vertoont. Hierbij zijn A, B en C vaagverzamelingen in $[0,1]$. Het is intuïtief duidelijk dat $E(A,C) = 0$ moet gelden voor elke similariteitsmaat



Figuur 5.1: Tegenvoorbeeld T-transitiviteit

E (met $E \in \mathcal{F}(\mathcal{F}([0,1])^2)$). Als E een T-equivalentie is, volgt uit de T-transitiviteit dat dan $T(E(A,B), E(B,C)) = 0$ moet gelden. Als bovendien T een t-norm is zonder nuldelers, volgt

hieruit dat $E(A,B) = 0$ of $E(B,C) = 0$ moet gelden. Nochtans zouden we verwachten van een similariteitsmaat dat zowel $E(A,B) > 0$ als $E(B,C) > 0$ geldt. Een compatibiliteitsmaat vormt een alternatief om similariteit uit te drukken, waarbij T-transitiviteit niet geëist wordt

Definitie 81 (Compatibiliteitsmaat). *Een tolerantierelatie C in $\mathcal{F}(X)$ wordt een compatibiliteitsmaat genoemd als voldaan is aan*

$$(\forall (A,B) \in \mathcal{F}(X)^2)(C(A,B) = 0 \Leftrightarrow A \cap B = \emptyset)$$

Voor een compatibiliteitsmaat C kan het gebeuren dat $C(A,B) = 1$ geldt, waarbij $A \neq B$, m.a.w. waarbij $(\exists x \in X)(A(x) \neq B(x))$. Aangezien dit voor sommige toepassingen een ongewenste eigenschap is, wordt in [77] voorgesteld om de similariteit te bepalen op basis van zowel een T-equivalentie als van een compatibiliteitsmaat en de bekomen waarden op een gepaste manier te combineren.

5.3.2 Inclusie

Voor de relatie "specifieker dan" kunnen we gebruik maken van een vage uitbreiding van het begrip partiële ordening. In [10] wordt hiervoor de volgende definitie voorgesteld.

Definitie 82 (T-E-ordening). *Zij T een t-norm en E een T-equivalentie. Een T-E-ordening in een universum X is dan een vaagrelatie N in X waarvoor voldaan is aan:*

E-reflexiviteit $(\forall (x,y) \in X^2)(E(x,y) \leq N(x,y))$

T-E-antisymmetrie $(\forall (x,y) \in X^2)(T(N(x,y), N(y,x)) \leq E(x,y))$

T-transitiviteit $(\forall (x,y,z) \in X^3)(T(N(x,y), N(y,z)) \leq N(x,z))$

Zij I een implicator waarvoor voldaan is aan de residueringsvoorwaarde (3.5) m.b.t. een zekere t-norm, m.a.w. I is het rechterresidu van een t-norm waarvan de partiële afbeeldingen supmorfismen zijn. Wegens (3.6) is de T-E-antisymmetrie equivalent met

$$(\forall (x,y) \in X^2)(I(T(N(x,y), N(y,x)), E(x,y)) = 1)$$

wat het verband met de antisymmetrievoorwaarde van een partiële ordening duidelijk weergeeft. Merk op dat E-reflexiviteit de gewone reflexiviteit impliceert. Een vaagrelatie in $\mathcal{F}(X)$ die voor vaagverzamelingen A en B weergeeft in welke mate A bevat is in B, wordt meestal een inclusiemaat genoemd. Voor scherpe verzamelingen in een universum X is bij definitie steeds voldaan aan

$$A \subseteq B \Leftrightarrow (\forall x \in X)(x \in A \Rightarrow x \in B)$$

De volgende uitbreiding naar vaagverzamelingen ligt dan voor de hand. Zij N_T de vaagrelatie in $\mathcal{F}(X)$, gedefinieerd voor A en B vaagverzamelingen in X als

$$N_T(A,B) = \inf_{x \in X} I_T(A(x), B(x)) \tag{5.3}$$

Hierbij is I_T het rechterresidu van een t-norm T.

Eigenschap 23. *Als de partiële afbeeldingen van T supmorfismen zijn, is N_T een T-E_T-ordening*

Bewijs. Zij I de met T corresponderende residuele implicator. Voor willekeurige vaagverzamelingen A en B in X hebben we

$$\begin{aligned} E_T(A,B) &= \inf_{x\in X} \min(I(A(x),B(x)), I(B(x),A(x))) \\ &= \min(\inf_{x\in X} I(A(x),B(x)), \inf_{x\in X} I(B(x),A(x))) \\ &\leq \inf_{x\in X} I(A(x),B(x)) \\ &= N_T(A,B) \end{aligned}$$

Voor de T-E_T-antisymmetrie vinden we voor willekeurige vaagverzamelingen A en B in X:

$$\begin{aligned} T(N_T(A,B),N_T(B,A)) &= T(\inf_{x\in X} I(A(x),B(x), \inf_{x\in X} I(B(x),A(x))) \\ &\leq \inf_{x\in X}\inf_{x'\in X} T(I(A(x),B(x)),I(B(x'),A(x'))) \\ &\leq \inf_{x\in X} T(I(A(x),B(x)),I(B(x),A(x))) \\ &= \inf_{x\in X} \varepsilon_T(A(x),B(x)) \\ &= E_T(A,B) \end{aligned}$$

Waarbij de eerste ongelijkheid volgt uit het stijgend zijn van T en eigenschap 1. Voor de T-transitiviteit vinden we tenslotte voor willekeurige vaagverzamelingen A, B en C in X:

$$\begin{aligned} T(N_T(A,B),N_T(B,C)) &= T(\inf_{x\in X} I(A(x),B(x)), \inf_{x\in X} I(B(x),C(x))) \\ &\leq \inf_{x\in X}\inf_{x'\in X} T(I(A(x),B(x)),I(B(x'),C(x'))) \\ &\leq \inf_{x\in X} T(I(A(x),B(x)),I(B(x),C(x))) \\ &\leq \inf_{x\in X} I(A(x),C(x)) \\ &= N_T(A,C) \end{aligned}$$

De eerste ongelijkheid volgt opnieuw uit het stijgend zijn van T en eigenschap 1, de laatste ongelijkheid volgt uit (3.12). □

Noch de vaagrelatie N_T, noch de vaagrelatie E_T zijn onmiddellijk bruikbaar om documenten te vergelijken. We zullen hier dieper op in gaan in paragraaf 5.4.

5.3.3 Bovenbenaderingen

De hybridisatie van vaagverzamelingen en ruwverzamelingen vormt een natuurlijke manier om verbanden tussen termen in rekening te brengen wanneer documenten door vaagverzamelingen van termen worden gerepresenteerd. Veronderstellen we dat $E^{\mathcal{T}}$ en $N^{\mathcal{T}}$ vaagrelaties in $\mathcal{T}$ zijn die respectievelijk de similariteit tussen termen en de relatie "specifieker dan" modelleren. Deze vaagrelaties kunnen automatisch geconstrueerd worden uitgaande van de document-termrelatie R, zoals we verder zullen zien, of kunnen handmatig geconstrueerd worden. De bovenbenadering $\overline{E^{\mathcal{T}}}(dR)$, met T een t-norm, stelt de vaagverzameling voor van alle termen die mogelijk relevant zijn voor het document d uit $\mathcal{D}$. Deze mogelijk relevante termen zijn in dit geval de termen die verwant zijn met termen uit het document, zoals bijvoorbeeld

synoniemen van termen die voorkomen in het document. Afhankelijk van de semantiek van $E^{\mathcal{T}}$ kunnen bijvoorbeeld ook antoniemen van termen die voorkomen in het document relevant beschouwd worden. Wanneer gebruik gemaakt wordt van de bovenbenadering $\overline{N^{\mathcal{T}}}(dR)$, met T een t-norm, zijn de mogelijk relevante termen alle termen die specifieker zijn dan een term die voorkomt in het document d. Door gebruik te maken van onderbenaderingen zouden in theorie de essentiële termen van een document geïdentificeerd kunnen worden. In de praktijk is deze onderbenadering, door de aanwezigheid van het infimum in de definitie, nagenoeg altijd de lege verzameling. In [38] worden twee clusteringsalgoritmen voorgesteld die gebruik maken van bovenbenaderingen in de benaderingsruimte $(\mathcal{T}, I_\theta)$. De (scherpe) ononderscheidbaarheidsrelatie I_θ wordt hierbij voor termen t_1 en t_2 in $\mathcal{T}$ gedefinieerd als

$$(t_1, t_2) \in I_\theta \Leftrightarrow E^{\mathcal{T}}(t_1, t_2) \geq \theta$$

Tabel 5.1 vat de belangrijkste verschillen samen tussen de aanpak m.b.v. vaagverzamelingen en de vectorgebaseerde aanpak.

	Vectoren	Vaagverzamelingen
Documentrepresentatie	vector i.d. documentruimte	vaagverzameling in $\mathcal{T}$
Similariteit	vb. cosinussimilariteit	vb. E_T
"Specifieker dan"	-	vb. N_T
Verbanden tussen termen	dimensiereductie	bovenbenadering

Tabel 5.1: Verschillen tussen de vector- en vaagverzamelinggebaseerde aanpak

5.4 Documenten als formele vaagconcepten

Zij $\mathcal{D} = \{d_1, d_2 \ldots, d_n\}$ nog steeds een eindig, niet-ledig universum van documenten, $\mathcal{T} = \{t_1, t_2, \ldots, t_m\}$ de eindige verzameling van termen die voorkomen in de documenten uit $\mathcal{D}$ en R de vaagrelatie van $\mathcal{D}$ naar $\mathcal{T}$ zodat $R(d, t)$ voor d in $\mathcal{D}$ en t in $\mathcal{T}$, de mate aangeeft waarin t relevant is voor d. Het drietal $(\mathcal{D}, \mathcal{T}, R)$ is dan een formele vaagcontext. Wegens eigenschap 17 kunnen we met een willekeurige vaagverzameling B in $\mathcal{T}$ het vaagconcept $(B^{\twoheadleftarrow}, B^{\twoheadrightarrow\twoheadleftarrow})$ laten corresponderen en kunnen we met een willekeurige vaagverzameling A in $\mathcal{D}$ het vaagconcept $(A^{\twoheadrightarrow\twoheadleftarrow}, A^{\twoheadrightarrow})$ laten corresponderen. Beschouwen we bijvoorbeeld het speciale geval waarbij A de scherpe deelverzameling van $\mathcal{D}$ is die enkel het document d bevat. In deze paragraaf zullen we steeds veronderstellen dat I een implicator is, T een t-norm is en dat I en T voldoen aan de residueringsvoorwaarde (3.5). Voor een willekeurige term t geldt er

$$\begin{aligned}
A^{\twoheadrightarrow}(t) &= \inf_{d' \in \mathcal{D}} I(A(d'), R(d', t)) \\
&= \min(\inf_{d' \in \mathcal{D} \setminus \{d\}} I(A(d'), R(d', t)), I(A(d), R(d, t))) \\
&= \min(\inf_{d' \in \mathcal{D} \setminus \{d\}} I(0, R(d', t)), I(1, R(d, t))) \\
&= \min(\inf_{d' \in \mathcal{D} \setminus \{d\}} 1, R(d, t)) \\
&= R(d, t)
\end{aligned}$$

waarbij we gebruik hebben gemaakt van $I(0,x)=1$, voor alle x in $[0,1]$ en van het feit dat I een randimplicator is. Meer algemeen kunnen we $A^{\rightarrow}$, met A een willekeurige vaagverzameling in $\mathcal{D}$, interpreteren als de vaagverzameling van termen die relevant zijn voor alle documenten uit A. Verder hebben we voor een willekeurig document d'

$$\begin{aligned} A^{\rightarrow\leftarrow}(d') &= \inf_{t'\in\mathcal{T}} I(A^{\rightarrow}(t'), R(d',t')) \\ &= \inf_{t'\in\mathcal{T}} I(R(d,t'), R(d',t')) \\ &= d(R \triangleleft R^{-1})d' = N_T(dR, d'R) \end{aligned}$$

Hierbij wordt N_T gedefinieerd zoals in (5.3). Meer algemeen kunnen we $A^{\rightarrow\leftarrow}$, met A een willekeurige vaagverzameling in $\mathcal{D}$, interpreteren als de vaagverzameling van documenten die minstens alle termen bevatten die relevant zijn voor alle documenten uit A.

Met elk document, correspondeert er dus een vaagconcept (A_d, B_d) met $A_d = d(R \triangleleft R^{-1})$ een vaagverzameling die alle documenten bevat die algemener zijn m.b.t. de vaagrelatie N_T dan d en waarbij de vaagverzameling $B_d = dR$ de representatie is van het document d. Relaties tussen termen kunnen gemakkelijk in rekening worden gebracht door gebruik te maken van de bovenbenadering van de context $(\mathcal{D}, \mathcal{T}, R)$ m.b.t. een relatie $E^{\mathcal{T}}$ die de similariteit van termen modelleert of m.b.t. een relatie $N^{\mathcal{T}}$ die het begrip "specifieker dan" voor termen modelleert.

5.4.1 Similariteit

Door met documenten concepten te laten corresponderen, dringt volgende definitie voor de similariteit van documenten zich op: twee documenten zijn gelijk in de mate dat ze door geen enkel vaagconcept gescheiden worden. We definiëren de vaagrelatie $E_1^{\mathcal{D}}$ in $\mathcal{D}$ voor documenten d en e als

$$E_1^{\mathcal{D}}(d,e) = \inf_{(A,B)\in\mathcal{B}(\mathcal{D},\mathcal{T},R)} \varepsilon_T(A(d), A(e))$$

Het overlopen van alle concepten voor het bepalen van het infimum is praktisch niet haalbaar. In [6] wordt er echter aangetoond dat voldaan is aan

$$\inf_{(A,B)\in\mathcal{B}(\mathcal{D},\mathcal{T},R)} \varepsilon_T(A(d), A(e)) = \inf_{t\in\mathcal{T}} \varepsilon_T(R(d,t), R(e,t)) \tag{5.4}$$

We hebben m.a.w. dat $E_1^{\mathcal{D}}(d,e) = E_T(dR, eR)$ waarbij E_T gegeven wordt door (5.2). In [6] wordt bovendien aangetoond dat $E_1^{\mathcal{D}}$ de grootste T-equivalentie is waarvoor voldaan is aan

$$(\forall (A,B)\in\mathcal{B}(\mathcal{D},\mathcal{T},R))(\forall (d,e)\in\mathcal{D}\times\mathcal{D})(T(A(d), E_1^{\mathcal{D}}(d,e)) \leq A(e)) \tag{5.5}$$

Aangezien $E_1^{\mathcal{D}}$ een T-equivalentierelatie is, is $(\mathcal{D}, E_1^{\mathcal{D}})$ een V-ruimte in de geresidueerde tralie $([0,1], \leq, T, I, 1)$. Uit (3.12) volgt dat ook $([0,1], I)$ een V-ruimte is in deze geresidueerde tralie. Wegens de residueringsvoorwaarde (3.5) en de commutativiteit van T, volgt uit (5.5) voor elk concept (A,B) dat A een V-map is van de V-ruimte $(\mathcal{D}, E_1^{\mathcal{D}})$ in de V-ruimte $([0,1], I)$, m.a.w. een V-predikaat in $(\mathcal{D}, E_1^{\mathcal{D}})$.

We zullen ons om de similariteit tussen documenten te bepalen, baseren op (5.4). Helaas houdt (5.4) enkel rekening met de slechtste term. We zullen daarom het infimum in (5.4) vervangen door een gewogen som. Zij $E_2^{\mathcal{D}}$ de vaagrelatie in $\mathcal{D}$, gedefinieerd voor d en e in $\mathcal{D}$

als

$$E_2^{\mathcal{D}}(d,e) = \frac{\sum_{i=1}^{m} w_i \cdot \varepsilon_T(R(d,t_i), R(e,t_i))}{\sum_{i=1}^{m} w_i}$$

Hierbij zijn $w_1, w_2, \ldots, w_m$ reële getallen in $[0,1]$, zodat w_i ($i \in \{1,2\ldots,m\}$) het belang aangeeft van de term t_i voor het bepalen van de similariteit tussen de documenten d en e. We onderstellen steeds $\sum_{i=1}^{m} w_i > 0$. Kiezen we bijvoorbeeld $w_i = \max(R(d,t_i), R(e,t_i))$ ($i \in \{1,2,\ldots,m\}$), dan wordt bovenstaande definitie voor $I = I_P$

$$\begin{aligned} E_2^{\mathcal{D}}(d,e) &= \frac{\sum_{i=1}^{m} \max(R(d,t_i), R(e,t_i)) \cdot \varepsilon_{T_P}(R(d,t_i), R(e,t_i))}{\sum_{i=1}^{m} \max(R(d,t_i), R(e,t_i))} \\ &= \frac{\sum_{i=1}^{m} \min(R(d,t_i), R(e,t_i))}{\sum_{i=1}^{m} \max(R(d,t_i), R(e,t_i))} \\ &= sim_{jac}(dR, eR) \end{aligned}$$

Er werd gebruik gemaakt van de volgende hulpeigenschap:

Hulpeigenschap 1. *Voor a en b in $[0,1]$ is voldaan aan*

$$\max(a,b)\varepsilon_P(a,b) = \max(a,b)\min(I_P(a,b), I_P(b,a)) = \min(a,b)$$

Bewijs. Voor $a < b$ geldt

$$\max(a,b) \cdot \min(I_P(a,b), I_P(b,a)) = b \cdot \min(1, \frac{a}{b}) = b \cdot \frac{a}{b} = a = \min(a,b)$$

Voor $b < a$ hebben we

$$\max(a,b) \cdot \min(I_P(a,b), I_P(b,a)) = a \cdot \min(\frac{b}{a}, 1) = a \cdot \frac{b}{a} = b = \min(a,b)$$

Voor $a = b$ tenslotte geldt

$$\max(a,b) \cdot \min(I_P(a,b), I_P(b,a)) = \max(a,b) \cdot \min(1,1) = \max(a,b) = \min(a,b)$$

□

De similariteitsmaat sim_{jac} is bekend als de gewogen Jaccard coëfficiënt en wordt algemeen voor vaagverzamelingen A en B in een eindig universum X gegeven door

$$sim_{jac}(A,B) = \frac{\sum_{x \in X} \min(A(x), B(x))}{\sum_{x \in X} \max(A(x), B(x))}$$

Eigenschap 24. *De gewogen Jaccard coëfficiënt sim_{jac} is een compatibiliteitsmaat*

Bewijs. De reflexiviteit van sim_{jac} volgt uit $\min(a,a) = \max(a,a) = a$ voor alle a in $\mathbb{R}$. De symmetrie van sim_{jac} volgt uit de symmetrie van min en max. Bovendien hebben we voor vaagverzamelingen A en B in een universum X

$$\begin{aligned} sim_{jac}(A,B) = 0 &\Leftrightarrow \frac{\sum_{x\in X} \min(A(x),B(x))}{\sum_{x\in X} \max(A(x),B(x))} = 0 \\ &\Leftrightarrow \sum_{x\in X} \min(A(x),B(x)) = 0 \\ &\Leftrightarrow (\forall x \in X)(\min(A(x),B(x)) = 0) \\ &\Leftrightarrow (\forall x \in X)((A\cap B)(x) = 0) \\ &\Leftrightarrow A \cap B = \emptyset \end{aligned}$$

□

Eigenschap 25. *[19] De gewogen Jaccard coëfficiënt sim_{jac} is T_W transitief*

We kunnen sim_{jac} dus gebruiken voor het vaagmieralgoritme.

5.4.2 Inclusie

We beschouwen een document d specifieker dan een document e in de mate dat e voorkomt in de eerste component van alle vaagconcepten waarin d voorkomt in de eerste component. We definiëren de vaagrelatie $N_1^{\mathcal{D}}$ in $\mathcal{D}$ voor d en e in $\mathcal{D}$ als

$$N_1^{\mathcal{D}}(d,e) = \inf_{(A,B)\in\mathcal{B}(\mathcal{D},\mathcal{T},R)} I(A(d),A(e))$$

Volledig analoog aan het bewijs van (5.4) kunnen we bewijzen dat

$$N_1^{\mathcal{D}}(d,e) = \inf_{t\in T} I(R(d,t),R(e,t)) \tag{5.6}$$

We hebben m.a.w. dat $N_1^{\mathcal{D}}(d,e) = N_T(dR,eR)$ waarbij N_T gegeven wordt door (5.3). Hieruit volgt dat $N_1^{\mathcal{D}}$ een $T - E_1^{\mathcal{D}}$-ordening is.

We hebben bovendien dat $N_1^{\mathcal{D}}$ de grootste vaagrelatie in $\mathcal{F}(\mathcal{D})$ is waarvoor voldaan is aan

$$(\forall (A,B)\in\mathcal{B}(\mathcal{D},\mathcal{T},R))(\forall (d,e)\in\mathcal{D}\times\mathcal{D})(T(A(d),N_1^{\mathcal{D}}(d,e)) \le A(e)) \tag{5.7}$$

Wegens de residueringsvoorwaarde (3.5) en de commutativiteit van T hebben we immers dat (5.7) equivalent is met

$$(\forall (A,B)\in\mathcal{B}(\mathcal{D},\mathcal{T},R))(\forall (d,e)\in\mathcal{D}\times\mathcal{D})(N_1^{\mathcal{D}}(d,e) \le I(A(d),A(e)))$$

waaraan voldaan is, aangezien het infimum een ondergrens is. Stel bovendien dat ook de vaagrelatie N' in $\mathcal{F}(\mathcal{D})$ aan (5.7) voldoet, dan volgt uit het feit dat het infimum de grootste ondergrens is voor documenten d en e dat

$$N'(d,e) \le \inf_{(A,B)\in\mathcal{B}(\mathcal{D},\mathcal{T},R)} I(A(d),A(e)) = N_1^{\mathcal{D}}(d,e)$$

Merk op dat uit (5.6) volgt dat $N_1^{\mathcal{D}}$ de metriek voor V-predikaten in $(\mathcal{D}, E_1^{\mathcal{D}})$ is, die gegeven wordt door (3.2).

Opnieuw kunnen we het infimum vervangen door een gewogen som. Zij $N_2^{\mathcal{D}}$ de vaagrelatie in $\mathcal{D}$, gedefinieerd voor d en e in $\mathcal{D}$ als

$$N_2^{\mathcal{D}}(d,e) = \frac{\sum_{i=1}^{m} w_i \cdot I(R(d,t_i), R(e,t_i))}{\sum_{i=1}^{m} w_i}$$

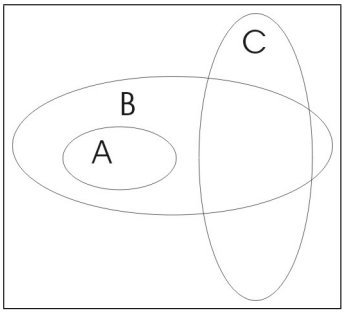
Hierbij zijn $w_1, w_2, \ldots, w_m$ reële getallen in $[0,1]$, zodat w_i ($i \in \{1,2\ldots,m\}$) het belang aangeeft van de term t_i voor het bepalen van de mate waarin d specifieker is dan e. We onderstellen hierbij steeds $\sum_{i=1}^{m} w_i > 0$. Het belang van een term hangt nu echter enkel af van de mate waarin deze term aanwezig is in het document d. Termen uit e die niet in d zitten, zijn hier immers niet relevant. Voor $w_i = R(d,t_i)$ ($i \in \{1,2\ldots,m\}$) en $I = I_P$ krijgen we

$$\begin{aligned}
N_2^{\mathcal{D}}(d,e) &= \frac{\sum_{i=1}^{m} R(d,t_i) I_P(R(d,t_i), R(e,t_i))}{\sum_{i=1}^{m} R(d,t_i)} \\
&= \frac{\sum_{R(d,t_i)\leq R(e,t_i)} R(d,t_i) I_P(R(d,t_i), R(e,t_i))}{\sum_{i=1}^{m} R(d,t_i)} \\
&\quad + \frac{\sum_{R(d,t_i)> R(e,t_i)} R(d,t_i) I_P(R(d,t_i), R(e,t_i))}{\sum_{i=1}^{m} R(d,t_i)} \\
&= \frac{\sum_{R(d,t_i)\leq R(e,t_i)} R(d,t_i)\cdot 1 + \sum_{R(d,t_i)> R(e,t_i)} R(d,t_i)\frac{R(e,t_i)}{R(d,t_i)}}{\sum_{i=1}^{m} R(d,t_i)} \\
&= \frac{\sum_{R(d,t_i)\leq R(e,t_i)} R(d,t_i) + \sum_{R(d,t_i)> R(e,t_i)} R(e,t_i)}{\sum_{i=1}^{m} R(d,t_i)} \\
&= \frac{\sum_{i=1}^{m} \min(R(d,t_i), R(e,t_i))}{\sum_{i=1}^{m} R(d,t_i)}
\end{aligned}$$

Uit het volgende tegenvoorbeeld volgt echter dat de T-transitiviteit van $N_2^{\mathcal{D}}$ voor geen enkele t-norm T kan gegarandeerd worden. Zij $\mathcal{D} = \{d_1, d_2, d_3\}$, $\mathcal{T} = \{t_1, t_2, t_3\}$ en zij R gedefinieerd door $R(d_1, t_1) = R(d_1, t_3) = R(d_3, t_2) = 0$ en $R(d_1, t_2) = R(d_2, t_1) = R(d_2, t_2) = R(d_2, t_3) = R(d_3, t_1) = R(d_3, t_3) = 1$, dan hebben we

$$T(N_2^{\mathcal{D}}(d_1, d_2), N_2^{\mathcal{D}}(d_2, d_3)) = T(1, \frac{2}{3}) = \frac{2}{3} > 0 = N_2^{\mathcal{D}}(d_1, d_3)$$

Dit wordt eveneens geïllustreerd in figuur 5.2. Voor een inclusiemaat N kunnen we verwachten dat $N(A, B) = 1$ en $N(A, C) = 0$. Hieruit volgt dat N niet T-transitief kan zijn voor een zekere t-norm T van zodra $N(B, C) > 0$. Dit betekent dus, wegens de onderstelling i.v.m. de



Figuur 5.2: Tegenvoorbeeld T-transitiviteit inclusie

T_W-transitiviteit, dat we in principe geen inclusiemaat kunnen gebruiken om documenten te clusteren met het vaagmieralgoritme. De T_W-transitiviteit werd ingevoerd om te kunnen garanderen dat de mieren niet bedrogen uitkomen bij het neerleggen van een hoop op een andere hoop. We proberen nu na te gaan of we een bovengrens kunnen vastleggen op de maximale fout die kan bekomen worden in het vaagmieralgoritme door gebruik te maken van $N_2^{\mathcal{D}}$. We beschouwen vooreerst de volgende hulpeigenschap.

Hulpeigenschap 2. *Voor x, y en z in $[0, 1]$ is steeds voldaan aan*

$$\min(x, y) - \min(x, z) \geq \min(y, z) - z$$

Bewijs. We bewijzen het gestelde m.b.v. de methode v.h. gevallenonderzoek.

- Voor $x \leq y \leq z$: hebben we $x - x \geq y - z \Leftrightarrow z \geq y$
- Voor $x \leq z \leq y$: hebben we $x - x \geq z - z \Leftrightarrow 0 \geq 0$
- Voor $y \leq x \leq z$: hebben we $y - x \geq y - z \Leftrightarrow z \geq x$
- Voor $y \leq z \leq x$: hebben we $y - z \geq y - z$
- Voor $z \leq x \leq y$: hebben we $x - z \geq z - z \Leftrightarrow x \geq z$
- Voor $z \leq y \leq x$: hebben we $y - z \geq z - z \Leftrightarrow y \geq z$

□

Eigenschap 26. *Voor d, e en f in $\mathcal{D}$ is voldaan aan*

$$N_2^{\mathcal{D}}(d,e) + N_2^{\mathcal{D}}(e,f) - \max\left(1, \frac{\sum_{t\in\mathcal{T}} R(e,t)}{\sum_{t\in\mathcal{T}} R(d,t)}\right) \leq N_2^{\mathcal{D}}(d,f)$$

Bewijs. Voor de eenvoud in notatie stellen we $a_t = R(d,t)$, $b_t = R(e,t)$ en $c_t = R(f,t)$ voor t in $\mathcal{T}$. Het gestelde wordt dan gegeven door

$$\frac{\sum_{t\in\mathcal{T}} \min(a_t,b_t)}{\sum_{t\in\mathcal{T}} a_t} + \frac{\sum_{t\in\mathcal{T}} \min(b_t,c_t)}{\sum_{t\in\mathcal{T}} b_t} - \max(1, \frac{\sum_{t\in\mathcal{T}} b_t}{\sum_{t\in\mathcal{T}} a_t}) \leq \frac{\sum_{t\in\mathcal{T}} \min(a_t,c_t)}{\sum_{t\in\mathcal{T}} a_t} \tag{5.8}$$

- Onderstellen we nu vooreerst dat $\sum_{t\in\mathcal{T}} a_t \geq \sum_{t\in\mathcal{T}} b_t$, m.a.w. $\max\left(1, \frac{\sum_{t\in\mathcal{T}} b_t}{\sum_{t\in\mathcal{T}} a_t}\right) = 1$. We hebben dan dat (5.8) equivalent is met

  $$\sum_{t\in\mathcal{T}} a_t \sum_{t\in\mathcal{T}} (\min(b_t,c_t) - b_t) \leq \sum_{t\in\mathcal{T}} b_t \sum_{t\in\mathcal{T}} (\min(a_t,c_t) - \min(a_t,b_t))$$

  Nu is wegens hulpeigenschap 2 voldaan aan

  $$\sum_{t\in\mathcal{T}} \min(b_t,c_t) - b_t \leq \sum_{t\in\mathcal{T}} (\min(a_t,c_t) - \min(a_t,b_t))$$

  En dus, aangezien $\sum_{t\in\mathcal{T}} b_t > 0$

  $$\sum_{t\in\mathcal{T}} b_t \sum_{t\in\mathcal{T}} (\min(b_t,c_t) - b_t) \leq \sum_{t\in\mathcal{T}} b_t \sum_{t\in\mathcal{T}} (\min(a_t,c_t) - \min(a_t,b_t)) \tag{5.9}$$

  Uit $\sum_{t\in\mathcal{T}} \min(b_t,c_t) - b_t \leq 0$ en de onderstelling $\sum_{t\in\mathcal{T}} a_t \geq \sum_{t\in\mathcal{T}} b_t$ volgt

  $$\sum_{t\in\mathcal{T}} a_t \sum_{t\in\mathcal{T}} (\min(b_t,c_t) - b_t) \leq \sum_{t\in\mathcal{T}} b_t \sum_{t\in\mathcal{T}} (\min(b_t,c_t) - b_t) \tag{5.10}$$

  Wegens de transitiviteit van $\leq$ volgt uit (5.9) en (5.10) ten slotte het gestelde

- Zij nu $\sum_{t\in\mathcal{T}} a_t < \sum_{t\in\mathcal{T}} b_t$, m.a.w. $\max\left(1, \frac{\sum_{t\in\mathcal{T}} b_t}{\sum_{t\in\mathcal{T}} a_t}\right) = \frac{\sum_{t\in\mathcal{T}} b_t}{\sum_{t\in\mathcal{T}} a_t}$. We hebben dan dat (5.8) equivalent is met

  $$\sum_{t\in\mathcal{T}} a_t \sum_{t\in\mathcal{T}} \min(b_t,c_t) \leq \sum_{t\in\mathcal{T}} b_t \sum_{t\in\mathcal{T}} (\min(a_t,c_t) - \min(a_t,b_t) + b_t)$$

  Uit hulpeigenschap 2 volgt

  $$\sum_{t\in\mathcal{T}} \min(b_t,c_t) \leq \sum_{t\in\mathcal{T}} (\min(a_t,c_t) - \min(a_t,b_t) + b_t)$$

  wat wegens $\sum_{t\in\mathcal{T}} b_t > 0$ equivalent is met

  $$\sum_{t\in\mathcal{T}} b_t \sum_{t\in\mathcal{T}} \min(b_t,c_t) \leq \sum_{t\in\mathcal{T}} b_t \sum_{t\in\mathcal{T}} (\min(a_t,c_t) - \min(a_t,b_t) + b_t) \tag{5.11}$$

  Uit $\sum_{t\in\mathcal{T}} \min(b_t,c_t) \geq 0$ en de onderstelling $\sum_{t\in\mathcal{T}} a_t < \sum_{t\in\mathcal{T}} b_t$ volgt

  $$\sum_{t\in\mathcal{T}} a_t \sum_{t\in\mathcal{T}} \min(b_t,c_t) \leq \sum_{t\in\mathcal{T}} b_t \sum_{t\in\mathcal{T}} \min(b_t,c_t) \tag{5.12}$$

  Waaruit wegens (5.11), (5.12) en de transitiviteit van $\leq$, opnieuw het gestelde volgt.

□

5.5 Vergelijken van termen

5.5.1 Termrelaties

Zoals we reeds aanhaalden, laat het begrip bovenbenadering ons toe om een gegeven relatie tussen termen te betrekken bij het vergelijken van documenten. Hoe deze relatie tot stand komt hebben we tot dusver in het midden gelaten. Een voor de hand liggende oplossing zou zijn om gebruik te maken van handgemaakte synoniemenwoordenboeken, meestal thesauri genoemd. Deze aanpak heeft echter een aantal belangrijke nadelen. Vooreerst is het handmatig opstellen van een dergelijke thesaurus een tijdrovende bezigheid. Bestaande thesauri, zoals WordNet[1], behandelen enkel algemene begrippen en zijn dus niet toepasbaar voor gespecialiseerde toepassingen. Andere thesauri, zoals de UMLS Metathesaurus[2], behandelen enkel een specifiek domein. Bovendien kunnen bepaalde relaties tussen termen afhankelijk zijn van de context waarin ze gebruikt worden. Wanneer we zoekresultaten wensen te clusteren, zullen we gebruik moeten maken van relaties tussen termen uit gespecialiseerde domeinen die op voorhand niet gekend zijn. We bespreken in deze paragraaf hoe relaties tussen termen automatisch kunnen worden opgesteld. Voor A en B (scherpe) verzamelingen in $\mathcal{T}$ zullen we nagaan in welke mate elk document die de termen uit A bevat, ook de termen uit B bevat, we wensen m.a.w. de geldigheid na te gaan van de regel

Als een document de termen uit A bevat, dan bevat dit document de termen uit B

We zouden hiervoor, volledig analoog als voor documenten, formele conceptanalyse kunnen gebruiken. We bespreken in de volgende paragraaf het verband met vaagassociatieregels.

5.5.2 Vaagassociatieregels

Zij $\mathcal{U}$ een universum van objecten, $\mathcal{A}$ een universum van attributen en R een vaagrelatie van $\mathcal{U}$ naar $\mathcal{A}$ waarbij $R(u, a)$ voor u in $\mathcal{U}$ en a in $\mathcal{A}$ geïnterpreteerd wordt als de mate waarin u het attribuut a bezit. Zij $A = \{a_1, a_2, \ldots a_n\}$ en $B = \{b_1, b_2, \ldots, b_m\}$ (scherpe) verzamelingen in $\mathcal{U}$, een uitspraak van de vorm "Als een object de attributen uit A bezit, dan bezit dit object de attributen uit B", wordt een (vaag)associatieregel genoemd. We zullen deze uitspraak kortweg noteren als $A \Rightarrow B$. Scherpe associatieregels, m.a.w. associatieregels waarbij de incidentierelatie R scherp is, werden voor het eerst gebruikt voor de analyse van het koopgedrag om te bepalen welke producten in een winkel dicht bij elkaar geplaatst moeten worden. Objecten corresponderen in dit geval met transacties en attributen met aangekochte producten. Om de mate te bepalen waarin voldaan is aan een associatieregel $A \Rightarrow B$ zullen we gebruik maken van twee kwaliteitsmaten: de ondersteuning (Eng. support) en het vertrouwen (Eng. confidence). De ondersteuning *supp* weerspiegelt het aantal positieve voorbeelden van de associatieregel en wordt meestal gedefinieerd als [22]

$$supp(A \Rightarrow B) = \sum_{u \in \mathcal{U}} (D_A \cap_T D_B)(u)$$

[1] http://www.cogsci.princeton.edu/ wn/

[2] Unified Medical Language System: http://www.nlm.nih.gov/research/umls/umlsmain.html

Hierbij is T een t-norm en zijn D_A en D_B de vaagverzamelingen in $\mathcal{U}$ gedefinieerd voor u in $\mathcal{U}$ door

$$\begin{aligned} D_A(u) &= T(R(u,a_1), R(u,a_2), \ldots, R(u,a_n)) \\ D_B(u) &= T(R(u,b_1), R(u,b_2), \ldots, R(u,b_m)) \end{aligned}$$

Om het vertrouwen in een (vaag)associatieregel $A \Rightarrow B$ te bepalen worden in [22] volgende twee alternatieven voorgesteld

$$conf_1(A \Rightarrow B) = \frac{\sum_{u \in \mathcal{U}} (D_A \cap_T D_B)(u)}{\sum_{u \in \mathcal{U}} (D_A)(u)} \tag{5.13}$$

$$conf_2(A \Rightarrow B) = \frac{\sum_{u \in \mathcal{U}} (D_A \cap_T D_B)(u)}{\sum_{u \in \mathcal{U}} (D_A \cap_T D_B)(u) + \sum_{u \in \mathcal{U}} (D_A \cap_T co_N D_B)(u)} \tag{5.14}$$

Hierbij is T een t-norm, S een t-conorm en N een negator.

Wanneer we dit nu toepassen om afhankelijkheden tussen termen in een documentenverzameling te ontdekken, corresponderen documenten met objecten en termen met attributen. Stellen we bovendien $A = \{t\}$ en $B = \{t'\}$, met t en t' in $\mathcal{T}$, dan kunnen $conf(A \Rightarrow B)$ en $supp(A \Rightarrow B)$ benut worden om te bepalen in welke mate de term t specifieker is dan t'. We definiëren de vaagrelatie $N^{\mathcal{T}}$ in $\mathcal{T}$ voor termen t en t' als

$$N^{\mathcal{T}}(t,t') = \begin{cases} conf(\{t\} \Rightarrow \{t'\}) & \text{als } supp(\{t\} \Rightarrow \{t'\}) > k \\ 0 & \text{anders} \end{cases}$$

Hierbij wordt $conf$ gegeven door (5.13) of door (5.14) en is k een natuurlijke constante. Voor t en t' in $\mathcal{T}$ interpreteren we $N^{\mathcal{T}}(t,t')$ als de mate waarin de term t specifieker is dan de term t'. De similariteit tussen termen drukken we uit a.d.h.v de vaagrelatie $E^{\mathcal{T}}$ in $\mathcal{T}$ die voor t en t' in $\mathcal{T}$ gedefinieerd wordt als $E^{\mathcal{T}}(t,t') = T(N^{\mathcal{T}}(t,t'), N^{\mathcal{T}}(t',t))$, met T een zekere t-norm. Merk op dat similariteit hier niet onmiddellijk slaat op semantische verwantschap. Voor een Engelstalige documentenverzameling over Amerikaanse politiek zullen de termen “White” en “House” bijvoorbeeld een hoge similariteitswaarde krijgen.

Hoofdstuk 6

Clusteren van zoekresultaten

6.1 Zoekmachines

De enorme hoeveelheid beschikbare informatie op het web[1], maakt het zoeken naar en het organiseren van relevante informatie verre van triviaal. Zoekmachines, zoals Google[2], vormen de manier bij uitstek om de zoektocht naar informatie aan te vangen. Naast de gigantische omvang van het web, wordt het ontwerpen van zoekmachines bemoeilijkt door de grote diversiteit van de webpagina's. Deze systemen moeten bovendien bruikbaar zijn voor onervaren gebruikers [62]. De klassieke aanpak bestaat erin de gebruiker enkele zoektermen te laten opgeven en vervolgens een lijst te presenteren met webpagina's die relevant geacht worden. Webpagina's worden in deze lijst meestal voorgesteld a.d.h.v. hun titel, de url (uniform resource locator) van de pagina en een kort uittreksel van de pagina dat een snipper (snippet) genoemd wordt. Aangezien het aantal elementen in deze lijst dikwijls in de duizenden, of zelfs in de miljoenen, loopt, is een goede ordening cruciaal voor de bruikbaarheid van een dergelijk systeem. Bij het opstellen van zo'n ordening zal rekening moeten worden gehouden met webpagina's die speciaal werden opgesteld om het gebruikte algoritme te misleiden [62].

In het vectorruimtemodel kunnen de zoektermen worden voorgesteld in dezelfde vectorruimte als de documenten zelf. De vraag (query) van de gebruiker kan dan beschouwd worden als een document met als termen die zoektermen die door de gebruiker werden opgegeven. Een similariteitsmaat voor documenten, zoals de cosiunussimilariteit, kan dan gebruikt worden om de relevantie van elk document m.b.t. de gestelde vraag te bepalen. Deze methode is echter veel te beperkt om als dusdanig gebruikt te kunnen worden. Zo wordt bijvoorbeeld geen rekening gehouden met de populariteit van de webpagina en is de methode vatbaar voor manipulatie: de webpagina die het meest aantal keer de gevraagde zoektermen bevat, zou als eerste in de rangschikking terecht komen. Het Pagerank algoritme [62], definieert een totale ordening voor alle geïndexeerde webpagina's in een zoekmachine, gebaseerd op de populariteit van de pagina. Hierbij wordt gebruik gemaakt van de linkstructuur van het web. Zij B_u de verzameling webpagina's die een link naar een webpagina u bevatten. Zij N_v het aantal links op de webpagina v. Een mogelijke rangschikking wordt dan gedefinieerd door de volgende

[1] Doorheen dit hoofdstuk bedoelen we steeds het Wereldwijde Web (World Wide Web) wanneer we over het web spreken.

[2] http://www.google.be

recursieve vergelijking [62]:

$$R(u) = 1 - c + c \sum_{v \in B_u} \frac{R(v)}{N_v} \tag{6.1}$$

waarbij $R(u)$ het belang van de pagina u voorstelt en c een factor in $]0, 1[$ is die de verdampingsfactor genoemd wordt. Hoe groter de waarde van c, hoe groter de invloed van het aantal pagina's dat naar u verwijst. Merk op dat de noemer N_v in (6.1) steeds groter is dan 0, aangezien v ten minste een link naar u bevat. De rangschikking die door een zoekmachine teruggegeven wordt aan de gebruiker, wordt dan bekomen door de populariteit van een pagina op gepaste wijze te combineren met zijn relevantie voor de gestelde vraag.

Wanneer een gebruiker op zoek is naar een specifieke webpagina en bijvoorbeeld over de titel van deze pagina beschikt, leveren zoekmachines prima dienst. Dikwijls wordt door een gebruiker echter op zoek gegaan naar bepaalde informatie, zonder hierbij specifieke webpagina's op het oog te hebben. Zoekmachines geven de lijst van resultaten terug in vorm van webpagina's die typisch 10 resultaten bevatten. In [42] wordt vastgesteld dat meer dan de helft van de gebruikers enkel de eerste resultaatpagina bekijkt. Meer dan 80% van de gebruikers bekijkt ten hoogste de eerste drie resultaatpagina's. Nochtans is het belang van een webpagina inherent subjectief en afhankelijk van de voorkennis, interesses en doelstellingen van de gebruiker [62] en bestaat er dus niet steeds een rangschikking die voor alle gebruikers optimaal is. In [54] wordt een experiment beschreven waarbij het surfgedrag van acht ervaren webgebruikers werd geobserveerd. Bij de aanvang van het experiment werd met elke deelnemer een onderwerp afgesproken waarover deze nog nooit informatie op het web had opgezocht. De onderwerpen, variërend van "Good places in France for skiing" tot "Restaurants in Scandinavia with at least one star in the Michelin Guide", werden zodanig gekozen dat het vinden van relevante informatie niet triviaal was. Vervolgens werd aan de deelnemers gevraagd om gedurende een half uur op het web informatie op te zoeken over het afgesproken onderwerp. Hun surfgedrag werd hierbij gefilmd en de deelnemers werden nadien geconfronteerd met deze beelden. Nagenoeg alle deelnemers hanteerden geen specifieke strategie bij het zoeken naar informatie, maar lieten zich leiden door toeval en door visuele structuren die hen gepresenteerd werden. Bovendien werd vastgesteld dat veel deelnemers pas na lang vruchteloos zoeken overgingen tot geavanceerdere strategieën zoals het verfijnen van de zoektermen.

Op basis van logbestanden van verschillende zoekmachines werd ook in [3] en [43] vastgesteld dat weinig gebruikers hun zoektermen verfijnen en dat geavanceerde zoekmethoden die door zoekmachines meestal worden aangeboden, zoals het gebruik van booleaanse operatoren, niet of nauwelijks gebruikt worden. Ook technieken voor relevantiefeedback, zoals de mogelijkheid om gelijkaardige pagina's op te vragen, worden zelden gehanteerd [43]. Op basis van logbestanden van Google, AOL[3] en MSN[4], wordt in [31] vastgesteld dat het gebruik van booleaanse operatoren en frasen (m.a.w. letterlijke opeenvolgingen van termen, aangegeven d.m.v. aanhalingstekens), weinig invloed hebben op de ordening van de eerste 10 resultaten en vaak gedrag vertonen dat in strijd is met de intuïtie. Bovendien bevatte de eerste resultaatpagina in bijna 30% van de gevallen minder relevante resultaten wanneer gebruik gemaakt werd van booleaanse operatoren of frasen.

Deze bevindingen leiden tot de conclusie dat er, ondanks het feit dat de huidige zoekmachines in veel gevallen een adequate oplossing bieden, nog heel wat ruimte is voor verbetering, in het bijzonder m.b.t. de weergave van de resultaten. Methoden zoals relevantiefeedback en

[3] http://search.aol.com/aolcom/index.jsp
[4] http://search.msn.com/

geavanceerde zoekmogelijkheden, worden in de praktijk nauwelijks gebruikt. Een door mensen opgestelde onderverdeling in categorieën, zoals Yahoo[5] is onvoldoende schaalbaar om een substantieel deel van het web te kunnen omvatten. We zullen ons in dit hoofdstuk verder toeleggen op het door Cutting et al. voorgestelde idee [17] om een documentenverzameling te clusteren vooraleer deze aan de gebruiker te presenteren.

6.2 Algoritmen voor het clusteren van zoekresultaten

Het idee om een documentenverzameling te clusteren en deze clusters te presenteren in plaats van bijvoorbeeld een lijst van resultaten, werd los van de context van het web geïntroduceerd in het Scatter/Gather systeem van Cutting et al. [17]. Initieel krijgt een gebruiker van het Scatter/Gather systeem een aantal clusters te zien, die de volledige documentenverzameling weerspiegelen. De gebruiker kiest dan een aantal clusters, die samengevoegd worden en opnieuw worden geclusterd, zodat een clustering bekomen wordt van een deel van de documentenverzameling. Dit proces kan worden herhaald, tot de bekomen clusters specifiek genoeg zijn voor de doeleinden van de gebruiker. Het clusteren van de documenten moet hierbij bijzonder snel gebeuren, aangezien het systeem interactief gebruikt wordt. Agglomeratief hiërarchisch clusteren (AHC)[6] is wegens zijn kwadratische ondergrens voor de uitvoeringstijd bijgevolg niet bruikbaar. Anderzijds zijn partitiegebaseerde clusteringsalgoritmen, zoals het k-gemiddelde algoritme, typisch bijzonder gevoelig aan de initiële partitionering en dus als dusdanig evenmin bruikbaar.

Cutting et al. stellen hiertoe twee mogelijke alternatieven voor: buckshot en fractionation. Beide algoritmen maken gebruik van een (willekeurig) hiërarchisch clusteringsalgoritme met uitvoeringstijd kwadratisch in de grootte van de documentenverzameling, om de centra van de clusters te vinden. Alle documenten worden vervolgens toegevoegd aan de cluster die correspondeert met het dichtst gelegen centrum m.b.t. de gehanteerde dissimilariteitsmaat. Noem k het gewenste aantal clusters en n het aantal documenten in de documentenverzameling. Buckshot past eenvoudigweg het gekozen hiërarchisch algoritme toe op een deelverzameling van de documentenverzameling van grootte $\sqrt{kn}$. Fractionation verdeelt de documentverzameling op in een aantal groepen die emmers (buckets) genoemd worden, en past het gekozen hiërarchisch clusteringsalgoritme toe op alle emmers. Vervolgens worden de bekomen clusters als atomaire objecten beschouwd en wordt de procedure herhaald, tot (ten hoogste) k clusters overblijven. Nadat de documenten aan de centra zijn toegewezen kunnen nog een aantal verfijningen doorgevoerd worden zoals het iteratief toevoegen van de documenten aan de cluster met dichtst gelegen centrum, het samennemen van clusters waarvan de centra dichter bij elkaar gelegen zijn dan een opgegeven drempelwaarde en het opsplitsen van clusters die slechter scoren dan een opgegeven drempelwaarde voor een zeker coherentiecriterium.

Zamir en Etzioni onderzochten als eerste het clusteren van zoekresultaten in de context van het web [84]. Ze identificeerden volgende vereisten waaraan een algoritme om zoekresultaten te clusteren moet voldoen [85]:

Coherente clusters De clusters moeten zodanig geconstrueerd worden dat documenten die relevant zijn voor een bepaalde vraag afgezonderd worden van irrelevante documenten. Documenten kunnen bovendien verschillende onderwerpen behandelen. Het is daarom noodzakelijk dat documenten tot verschillende clusters tegelijk kunnen behoren.

[5] http://www.yahoo.com

[6] Voor een bespreking van AHC en andere klassieke clusteringsalgoritmen verwijzen we naar bijlage B

Doorbladerbaar op efficiënte wijze Het moet onmiddellijk, op basis van een korte beschrijving van de clusters, duidelijk zijn welke clusters relevant zijn en welke niet.

Snelheid Het aantal resultaten dat kan doorbladerd worden met het systeem moet een grootte-orde meer zijn dan het aantal resultaten dat met een lijstrepresentatie kan doorbladerd worden. Het moet dus mogelijk zijn om honderden tot duizenden documenten te clusteren in enkele seconden. De documenten moeten bovendien geclusterd worden op basis van de snippers die door de zoekmachine worden teruggegeven. Het is immers niet haalbaar om alle documenten volledig op te halen voor elke zoekopdracht.

Om hieraan tegemoet te komen, introduceerden Zamir en Etzioni in [84] het STC (Suffix Tree Clustering) algoritme. Door gebruik te maken van een datastructuur die een suffixboom genoemd wordt, wordt in lineaire tijd bepaald welke snippers een gemeenschappelijke term of frase (letterlijke opeenvolging van termen) bevatten. Elke knoop van de suffixboom correspondeert met een term of een frase en definieert dus een documentenverzameling die een basiscluster genoemd wordt. Met elke basiscluster wordt een score geassocieerd die berekend wordt op basis van het aantal documenten in de basiscluster en de lengte van de geassocieerde frase. Vervolgens wordt een graaf opgesteld, waarbij de knopen de basisclusters zijn. Twee basisclusters b_1 en b_2 worden in deze graaf met een boog verbonden a.s.a.

$$\frac{|b_1 \cap b_2|}{\max(|b_1|, |b_2|)} > 0.5$$

Een cluster wordt dan gedefinieerd als de unie van een verzameling basisclusters B zodat er een pad bestaat in de geconstrueerde graaf tussen elke twee basisclusters uit B en zodat elke basiscluster waarnaar er een pad bestaat vanuit een basiscluster uit B, zelf ook in B bevat is. De score van de clusters wordt berekend op basis van de scores van de basisclusters die erin bevat zitten en hun overlap. De clusters worden gerangschikt volgens deze score en de beste 10 clusters worden weergegeven. De frasen van de geassocieerde basisclusters worden hierbij als beschrijving voor de cluster gebruikt.

In [86] wordt een algoritme voorgesteld voor het clusteren van zoekresultaten dat gebaseerd is op LSI en waarbij documenten tot verschillende clusters kunnen behoren in een zekere mate. In [84] wordt experimenteel voor o.a. STC, buckshot, fractionation en k-gemiddelde aangetoond dat het clusteren van snippers i.p.v. volledige documenten geen grote invloed heeft op de kwaliteit van het resultaat. Op basis van logbestanden wordt in [85] vastgesteld dat met Grouper, de STC-implementatie van Zamir en Etzioni, meer gebruikers ten minste één link volgen dan met de klassieke lijstvoorstelling. De bekendste implementatie van een algoritme voor het clusteren van zoekresultaten is ongetwijfeld Vivisimo[7]. Helaas werd het gebruikte algoritme geheim gehouden.

6.3 Vaagmieren voor het clusteren van zoekresultaten

Het buckshot en fractionation algoritme veronderstellen dat het aantal clusters op voorhand gekend is. Het is duidelijk dat een dergelijke veronderstelling in de context van het web niet zinvol is. Ook het STC algoritme maakt een gelijkaardige veronderstelling door steeds de 10 clusters weer te geven met de hoogste score. Het gevolg is dat dikwijls heel wat clusters

[7] http://www.vivisimo.com

weergegeven worden die niet betekenisvol zijn. We zullen in deze paragraaf nagaan hoe we het vaagmieralgoritme uit hoofdstuk 4 kunnen gebruiken om zoekresultaten te clusteren zonder deze veronderstelling. We zullen hierbij gebruik maken van de inclusiematen voor documenten en termen die we in het vorige hoofdstuk besproken hebben. In tegenstelling tot het leeuwendeel van de bestaande clusteringsalgoritmen voor documenten, zullen we dus geen gebruik maken van een symmetrische (vaag)relatie. Het gebruik van asymmetrische relaties voor het clusteren van documenten en termen, werd reeds voorgesteld in [52].

6.3.1 Opstellen van de term- en documentrelaties

In elk document komen termen voor die geen inhoudelijke informatie met zich meebrengen. Typische voorbeelden zijn lidwoorden, voegwoorden, hulpwerkwoorden, ... Dergelijke woorden worden stopwoorden genoemd en kunnen ervoor zorgen dat documenten die inhoudelijk niets met elkaar gemeen hebben toch een strikt positieve similariteitswaarde krijgen. Daarom worden stopwoorden dikwijls buiten beschouwing gelaten bij het bepalen van de similariteit. Lijsten met stopwoorden voor een bepaalde taal kunnen, gegeven een documentenverzameling met documenten in deze taal, gemakkelijk automatisch worden opgesteld aangezien ze typisch een hoge frequentie van voorkomen hebben. Het bestaan van meervoudsvormen, vervoegingen, verbuigingen, ... in een taal, kan er anderzijds voor zorgen dat de similariteit van documenten die, bijvoorbeeld op hun verbuiging na, gelijke termen bevatten, toch gelijk is aan nul. Een veel gebruikte oplossing bestaat erin om voor het bepalen van de similariteit, alle termen terug te brengen naar hun grammaticale stamvorm. Dit wordt gerealiseerd door een algoritme dat een stemmer genoemd wordt. In [51] wordt een variant besproken van de bekende Porter stemmer voor de Nederlandse taal. Verder worden woorden die andere karakters dan letters, zoals bijvoorbeeld cijfers, bevatten buiten beschouwing gelaten. Door termen weg te laten die in heel weinig documenten voorkomen, wordt een veel efficiëntere implementatie bekomen. In onze implementatie hebben we geen rekening gehouden met termen die in minder dan drie snippers voorkomen, met een significante daling van het totale aantal termen tot gevolg. Ten slotte worden ook de termen die als zoektermen door de gebruiker werden opgegeven buiten beschouwing gelaten aangezien we kunnen verwachten dat deze voor alle snippers relevant zullen zijn.

Zij $\mathcal{D}$ de verzameling snippers en $\mathcal{T}$ de verzameling van de stamvormen van de termen die voorkomen in ten minste drie snippers uit $\mathcal{D}$, geen stopwoorden zijn en niet als zoekterm werden opgegeven. De document-termrelatie R_1 is de vaagrelatie van $\mathcal{D}$ naar $\mathcal{T}$ die voor d in $\mathcal{D}$ en t in $\mathcal{T}$ gedefinieerd wordt als

$$R_1(d,t) = \begin{cases} 1 & \text{als } t \text{ de stamvorm is van een term die voorkomt in } d \\ 0 & \text{anders} \end{cases}$$

We kiezen m.a.w. voor een gewichtschema met binaire gewichten. Gewichtschema's op basis van het aantal voorkomens van termen, zoals het TF-model, zijn hier niet zinvol, gezien de korte lengte van de snippers. Het penaliseren van termen die in veel documenten voorkomen, zou hier bovendien een nadelig effect veroorzaken, aangezien de potentieel aanwezige (verborgen) clusterstructuur hierdoor deels vernietigd zou worden.

Zij $N_1^{\mathcal{T}}$ de vaagrelatie in $\mathcal{T}$ die voor t_1 en t_2 in $\mathcal{T}$ gedefinieerd wordt als

$$N_1^{\mathcal{T}}(t_1, t_2) = \frac{\sum_{d \in \mathcal{D}} \min(R_1(d, t_1), R_1(d, t_2))}{\sum_{d \in \mathcal{D}} R_1(d, t_1)}$$

We zullen de relatie "specifieker dan" voor termen modelleren a.d.h.v. de vaagrelatie $N^{\mathcal{T}}$ in $\mathcal{T}$, gedefinieerd voor t_1 en t_2 in $\mathcal{T}$ als

$$N^{\mathcal{T}}(t_1, t_2) = \begin{cases} N_1^{\mathcal{T}}(t_1, t_2) & \text{als } N_1^{\mathcal{T}}(t_1, t_2) \geq 0.3 \\ 0 & \text{anders} \end{cases}$$

Voor termen t_1 en t_2 waarvoor $N_1^{\mathcal{T}}(t_1, t_2) < 0.3$ kunnen we niet op een betrouwbare manier besluiten dat t_1 specifieker is dan t_2. We verliezen dus niet veel informatie door $N(t_1, t_2)$ gelijk aan nul te stellen. In een implementatie zullen vaagrelaties steeds voorgesteld worden als (ijle) matrices. De complexiteit van de implementatie zal dan evenredig zijn met het aantal niet-nulposities in deze matrices. Wanneer geen gebruik gemaakt zou worden van het ijle karakter van de optredende matrices, zou het opstellen van de vaagrelatie $N_1^{\mathcal{T}}$ alleen al een complexiteit hebben die kwadratisch is in het aantal termen. Het is m.a.w. belangrijk voor de efficiëntie van het algoritme dat voldoende nulposities bewaard blijven. Voor dergelijke, en andere, implementatie-aspecten verwijzen we naar bijlage C.

Definiëren we ten slotte $(\mathcal{D}, \mathcal{T}, R)$ als de bovenbenadering van de formele vaagcontext $(\mathcal{D}, \mathcal{T}, R_1)$ in de veralgemeende vaagbenaderingsruimte $(\mathcal{T}, N^{\mathcal{T}})$, m.a.w. zij R de vaagrelatie van $\mathcal{D}$ naar $\mathcal{T}$ die voor d in $\mathcal{D}$ en t in $\mathcal{T}$ gedefinieerd wordt door

$$R(d, t) = \sup_{t' \in \mathcal{T}} \min(R_1(d, t'), N^{\mathcal{T}}(t, t'))$$

waarbij het minimum als t-norm gebruikt werd. We modelleren de relatie "specifieker dan" als de vaagrelatie $N^{\mathcal{D}}$ in $\mathcal{D}$ die voor snippers d_1 en d_2 in $\mathcal{D}$ gedefinieerd wordt als

$$N^{\mathcal{D}}(d_1, d_2) = \frac{\sum_{t \in \mathcal{T}} \min(R(d_1, t), R(d_2, t))}{\sum_{t \in \mathcal{T}} R(d_1, t)}$$

Door gebruik te maken van R i.p.v. R_1 in de definitie van $N^{\mathcal{D}}$, kunnen snippers die geen gemeenschappelijke termen bevatten, toch in een van nul verschillende mate, verwant beschouwd worden. Wanneer alle termen die in d_1 voorkomen specifieker zijn dan een term die voorkomt in d_2, zal steeds $N^{\mathcal{D}}(d_1, d_2) = 1$ gelden, hoewel d_1 en d_2 in dit geval niet noodzakelijk gemeenschappelijke termen bevatten.

6.3.2 Aanpassingen aan het algoritme

Overlap

Het vaagmieralgoritme uit hoofdstuk 4 laat niet toe dat een object tot verschillende clusters tegelijk behoort. Een eenvoudige manier om hieraan tegemoet te komen, bestaat erin om na afloop van het algoritme enkel de centra van de hopen te beschouwen, en het lidmaatschap

van een object o in een cluster met centrum c te laten afhangen van de verwantschap tussen o en c. Om dit lidmaatschap te bepalen, baseren we ons op het vaag c-gemiddelde algoritme. Zij $c_1, c_2, \ldots, c_l$ de centra van de hopen die ten minste 4 objecten bevatten, na afloop van het vaagmieralgoritme. We beschouwen m.a.w. hopen met minder dan 4 objecten als ruis. De lidmaatschapsgraad $C_i(d)$ van een snipper d in $\mathcal{D}$ in de i^{de} cluster wordt gegeven door

$$C_i(d) = \begin{cases} \dfrac{1}{\displaystyle\sum_{j=1}^{l} \left(\dfrac{1 - N^{\mathcal{D}}(d, c_i)}{1 - N^{\mathcal{D}}(d, c_j)} \right)^{\frac{2}{M-1}}} & \text{Als } N^{\mathcal{D}}(d, c_j) < 1, \text{ voor alle } j \text{ in } \{1, 2, \ldots, l\} \\ 1 & \text{Als } N^{\mathcal{D}}(d, c_i) = 1 \\ 0 & \text{Anders} \end{cases} \tag{6.2}$$

hierbij is $M > 1$. Hoe hoger de waarde van M, hoe vager de clustering zal zijn. In de limiet wanneer M tot 1 nadert, heeft elke snipper een lidmaatschapsgraad 1 in de meest verwante cluster en lidmaatschapsgraad 0 in de andere clusters. We veronderstellen hierbij dat er geen verschillende clusters zijn die even verwant zijn als de meest verwante cluster. Wanneer $(\forall j \in \{1, 2, \ldots, i-1, i+1, \ldots, l\})(N^{\mathcal{D}}(d, c_j) < N^{\mathcal{D}}(d, c_i))$ hebben we immers

$$\lim_{M \underset{>}{\longrightarrow} 1} \sum_{j=1}^{l} \left(\frac{1 - N^{\mathcal{D}}(d, c_i)}{1 - N^{\mathcal{D}}(d, c_j)} \right)^{\frac{2}{M-1}} = 1 + \lim_{M \underset{>}{\longrightarrow} 1} \sum_{j \neq i} \left(\frac{1 - N^{\mathcal{D}}(d, c_i)}{1 - N^{\mathcal{D}}(d, c_j)} \right)^{\frac{2}{M-1}} = 1 \tag{6.3}$$

en dus $\lim\limits_{M \underset{>}{\longrightarrow} 1} C_i(d) = 1$. De breuken in de som van het rechterlid van (6.3) zijn immers strikt kleiner dan 1. Wanneer voor een zekere j_0 in $\{1, 2, \ldots, l\}$ geldt $N^{\mathcal{D}}(d, c_i) < N^{\mathcal{D}}(d, c_{j_0})$ krijgen we

$$\begin{aligned} & \lim_{M \underset{>}{\longrightarrow} 1} \sum_{j=1}^{l} \left(\frac{1 - N^{\mathcal{D}}(d, c_i)}{1 - N^{\mathcal{D}}(d, c_j)} \right)^{\frac{2}{M-1}} \\ = \; & \lim_{M \underset{>}{\longrightarrow} 1} \left(\frac{1 - N^{\mathcal{D}}(d, c_i)}{1 - N^{\mathcal{D}}(d, c_{j_0})} \right)^{\frac{2}{M-1}} + \lim_{M \underset{>}{\longrightarrow} 1} \sum_{j \neq j_0} \left(\frac{1 - N^{\mathcal{D}}(d, c_i)}{1 - N^{\mathcal{D}}(d, c_j)} \right)^{\frac{2}{M-1}} = +\infty \end{aligned}$$

en dus $\lim\limits_{M \underset{>}{\longrightarrow} 1} C_i(d) = 0$. We hebben immers

$$\begin{aligned} \lim_{M \underset{>}{\longrightarrow} 1} \left(\frac{1 - N^{\mathcal{D}}(d, c_i)}{1 - N^{\mathcal{D}}(d, c_{j_0})} \right)^{\frac{2}{M-1}} &= +\infty \\ \lim_{M \underset{>}{\longrightarrow} 1} \sum_{j \neq j_0} \left(\frac{1 - N^{\mathcal{D}}(d, c_i)}{1 - N^{\mathcal{D}}(d, c_j)} \right)^{\frac{2}{M-1}} &\in]0, +\infty[\end{aligned}$$

In de limiet wanneer M tot oneindig nadert, is de lidmaatschap van alle snippers in alle clusters gelijk aan $1/l$. In de implementatie hebben we gebruik gemaakt van $M = 1.3$. Zoals we besproken hebben in hoofdstuk 2 werd in [44] een gelijkaardige aanpak voorgesteld, waarbij het algoritme van Monmarché gecombineerd werd met het vaag c-gemiddelde algoritme. De motivatie is hier enigzins anders, aangezien onze voornaamste drijfveer niet het corrigeren van classificatiefouten is. Daarom worden ook geen verschillende iteraties van het vaag c-gemiddelde algoritme gehanteerd.

Bij de presentatie van de clusters, worden voor elke cluster C_i de snippers getoond waarvan de lidmaatschap in die cluster ten minste 0.3 is. De documenten die gepresenteerd worden, zijn dus m.a.w. de documenten die bevat zijn in $(C_i)_{0.3}$. Snippers die tot geen enkele cluster ten minste in de mate 0.30 behoren, worden ondergebracht in een afzonderlijke cluster met label "Andere". We bespreken verder hoe we de labels van de andere clusters kunnen bepalen.

Centrum van een hoop

Als centrum van een hoop kiezen we de snipper uit de hoop die het meest representatief is voor deze hoop, m.a.w. de meest algemene snipper uit de hoop. Deze snipper wordt de medoïde van de hoop genoemd. Een voor de hand liggende keuze zou zijn om als medoïde van een hoop H, de snipper m uit H te kiezen waarvoor

$$\sum_{h \in H} N^{\mathcal{D}}(h, m) \tag{6.4}$$

maximaal is. Helaas vereist het bepalen van een dergelijke medoïde een aantal bewerkingen dat kwadratisch is in de grootte van de hoop. Zelfs door gebruik te maken van optimalisaties m.b.t. het ijle karakter van de matrix die geassocieerd is met de vaagrelatie R kan het bepalen van de medoïde van een hoop op deze manier niet voldoende efficiënt geïmplementeerd worden. De voornaamste reden hiervoor is dat het aantal snippers h waarvoor $N^{\mathcal{D}}(h, m) = 0$ geldt, typisch redelijk klein zal zijn. Het volstaat immers dat de bovenbenaderingen van h en m één term gemeenschappelijk hebben opdat $N^{\mathcal{D}}(h, m) > 0$ zou gelden. We stellen daarom een efficiëntere definitie voor. Definiëren we de leiderwaarde $l(t)$ van een term t uit $\mathcal{T}$ als

$$l^{\mathcal{T}}(t) = \sum_{t' \in \mathcal{T}} N^{\mathcal{T}}(t', t) \tag{6.5}$$

Een gelijkaardige definitie voor leiderwaarde werd ook in [52] gebruikt. In tegenstelling tot (6.4), kan (6.5) wel redelijk efficiënt geïmplementeerd worden. Het aantal termen t' dat specifieker is dan een gegeven term t voor de vaagrelatie $N^{\mathcal{T}}$ zal immers typisch klein zijn. Bovendien moet de vaagrelatie $N^{\mathcal{T}}$, in tegenstelling tot de vaagrelatie $N^{\mathcal{D}}$, sowieso volledig bepaald worden. Bij het vergelijken van snippers tijdens de uitvoering van het algoritme, kunnen we immers verwachten dat elke snipper ten minste één maal aan bod komt. Voor elke snipper d zal de vaagverzameling dR bijgevolg ten minste één maal bepaald moeten worden. Voor elke term zullen de specifiekere termen dus ten minste één maal moeten bepaald worden. Merk bovendien op dat (6.5) voor elke term slechts één maal moet bepaald worden, in tegenstelling tot (6.4) die telkens opnieuw zou berekend moeten worden wanneer een hoop gewijzigd wordt.

De leiderwaarde $l^{\mathcal{D}}(d)$ van een snipper d in $\mathcal{D}$ definiëren we als

$$l^{\mathcal{D}}(d) = \sum_{t \in \mathcal{T}} R_1(d, t) \cdot l^{\mathcal{T}}(t) \tag{6.6}$$

Het aantal termen in elke snipper is naar boven begrensd door een constante. De som in (6.6) is dus voor een constant aantal termen strikt groter dan 0. Gegeven de leiderwaarden voor de termen, kan de leiderwaarde voor een snipper dus in constante tijd bepaald worden.

Transitiviteit

In hoofdstuk 4 hebben we ondersteld dat de similariteitsmaat die gebruikt wordt om objecten te vergelijken T_W-transitief is. Zoals we in paragraaf 5.4.2 hebben aangetoond, is hieraan niet voldaan voor $N^{\mathcal{D}}$. Voor een hoop L met centrum c_L en een hoop H met centrum c_H is in het algemene geval bijgevolg niet voldaan aan

$$T_W(avg(L), N^{\mathcal{D}}(c_L, c_H)) \leq \frac{1}{|L|} \sum_{l \in L} N^{\mathcal{D}}(l, c_H)$$

De mieren kunnen dus m.a.w. bedrogen uitkomen bij het neerleggen van een hoop L op een hoop H. Voor d_1, d_2 en d_3 in $\mathcal{D}$ hebben we wel wegens eigenschap 26

$$N^{\mathcal{D}}(d_1, d_2) + N^{\mathcal{D}}(d_2, d_3) \leq N^{\mathcal{D}}(d_1, d_3) + \max\left(1, \frac{\sum_{t \in \mathcal{T}} R(d_2, t)}{\sum_{t \in \mathcal{T}} R(d_1, t)}\right)$$

en dus

$$T_W(N^{\mathcal{D}}(d_1, d_2), N^{\mathcal{D}}(d_2, d_3)) \leq N^{\mathcal{D}}(d_1, d_3) + \max\left(0, \frac{\sum_{t \in \mathcal{T}} R(d_2, t) - R(d_1, t)}{\sum_{t \in \mathcal{T}} R(d_1, t)}\right)$$

Door het weglaten van termen die in minder dan drie documenten voorkomen, kan het gebeuren dat voor zekere d in $\mathcal{D}$, geldt dat $\sum_{t \in \mathcal{T}} R(d_1, t) = 0$. We zullen dergelijke snippers voor het clusteren buiten beschouwing laten en deze na afloop van het algoritme onderbrengen in de cluster "Andere". We kunnen bijgevolg verder veronderstellen dat $\sum_{t \in \mathcal{T}} R(d_1, t) > 0$, voor elke d in $\mathcal{D}$. We hebben verder voor een hoop H met centrum c_H en een hoop L met centrum c_L

$$\begin{aligned} T_W(avg(L), N^{\mathcal{D}}(c_L, c_H)) &= T_W(\frac{1}{|L|} \sum_{l \in L} N^{\mathcal{D}}(l, c_L), N^{\mathcal{D}}(c_L, c_H)) \\ &\leq \frac{1}{|L|} \sum_{l \in L} T_W(N^{\mathcal{D}}(l, c_L), N^{\mathcal{D}}(c_L, c_H)) \\ &\leq \frac{1}{|L|} \sum_{l \in L} \left(N^{\mathcal{D}}(l, c_H) + \max\left(0, \frac{\sum_{t \in \mathcal{T}} R(c_L, t) - R(l, t)}{\sum_{t \in \mathcal{T}} R(l, t)}\right)\right) \end{aligned}$$

Intuïtief kunnen we dit als volgt interpreteren: wanneer de som van de termgewichten van meeste snippers in een hoop niet veel kleiner is dan de som van de termgewichten van het centrum van die hoop, zullen de mieren dus niet al te bedrogen uitkomen. In hoeverre deze veronderstelling realistisch is, moet experimenteel worden vastgesteld. Enerzijds is de lengte van elke snipper naar boven begrensd door een constante. Anderzijds bevat het centrum van een hoop, bij definitie, meer algemene termen waardoor de bovenbenadering meer termen zal bevatten.

Een andere, ietwat minder efficiënte, mogelijkheid bestaat erin $\frac{1}{|L|} \sum_{l \in L} N^{\mathcal{D}}(l, c_H)$ exact te berekenen. We kunnen in dit geval de onderstelling omtrent de T_W-transitiviteit, zonder neveneffecten te introduceren, laten vallen. Het is dit laatste alternatief dat we in onze implementatie hebben gebruikt.

Bepalen van de labels

Om de clusters op een efficiënte manier te kunnen doorbladeren, moet elke cluster voorzien zijn van een korte accurate beschrijving. Een cluster kan beschouwd worden als een vaagverzameling in $\mathcal{D}$, waarbij de lidmaatschapsgraden gegeven worden door (6.2). Met een vaagverzameling C in $\mathcal{D}$, m.a.w. een cluster, kunnen we in de formele vaagcontext $(\mathcal{D}, \mathcal{T}, R_1)$ op natuurlijke manier de vaagverzameling B in $\mathcal{T}$, gedefinieerd voor t in $\mathcal{T}$ als

$$B(t) = C^{\rightarrow_R}(t) = \inf_{d \in \mathcal{D}} I(C(d), R_1(d,t)) = R^{\triangleleft}(C)(t)$$

laten corresponderen. Wanneer verschillende snippers in de mate 1 tot dezelfde cluster behoren en wanneer deze snippers bovendien geen enkele term gemeenschappelijk hebben, zal B de lege verzameling zijn. Wanneer we de bovenbenadering $(\mathcal{D}, \mathcal{T}, R_1)$ zouden beschouwen, worden specifieke termen bevoordeeld t.o.v. algemene termen, wat evenmin wenselijk is. Deze methode is dus in de praktijk niet bruikbaar.

In [17] wordt voorgesteld om met elke term de som van zijn gewichten in alle documenten of snippers van een cluster te laten corresponderen. Als alternatief wordt in [17] voorgesteld om enkel de som van de gewichten in de eerste m documenten te beschouwen, voor een zekere natuurlijke constante m. We beschouwen nu een kleine variant hierop. Zij C een cluster, voorgesteld als een vaagverzameling in $\mathcal{D}$ en S de vaagverzameling in $\mathcal{T}$, gedefinieerd voor t in $\mathcal{T}$ als

$$S(t) = \frac{1}{|\mathcal{D}|} \sum_{d \in \mathcal{D}} C(d) \cdot R_1(d,t)$$

De term die de cluster C best beschrijft, kan dan bekomen worden door defuzzificatie van de vaagverzameling S. Meer concreet zullen we C beschrijven door de term t waarvoor

$$S(t) = hgt(S) = \sup_{t \in \mathcal{T}} S(t) \tag{6.7}$$

Aangezien $\mathcal{T}$ eindig is, wordt dit supremum steeds bereikt. In het geval dat verschillende termen voldoen aan (6.7), wordt willekeurig één van deze termen gekozen.

Dikwijls komen bepaalde opeenvolgingen van termen steeds na elkaar voor in de snippers. Clusterbeschrijvingen die slechts één term uit een dergelijke opeenvolging bevatten, zijn heel wat minder duidelijk dan clusterbeschrijvingen die de volledige opeenvolging van termen bevatten. We zullen ons daarom baseren op de volgende definitie van compleetheid.

Definitie 83 (compleetheid). *[86] Zij D een verzameling documenten. Een opeenvolging van termen $t_1, \ldots, t_n$ wordt rechts-compleet, resp. links-compleet, genoemd in D als ten minste twee voorkomens van deze opeenvolging in D bestaan die gevolgd worden, resp. voorafgegaan worden, door een verschillende term. Een opeenvolging van termen die zowel links- als rechts-compleet is, wordt compleet genoemd.*

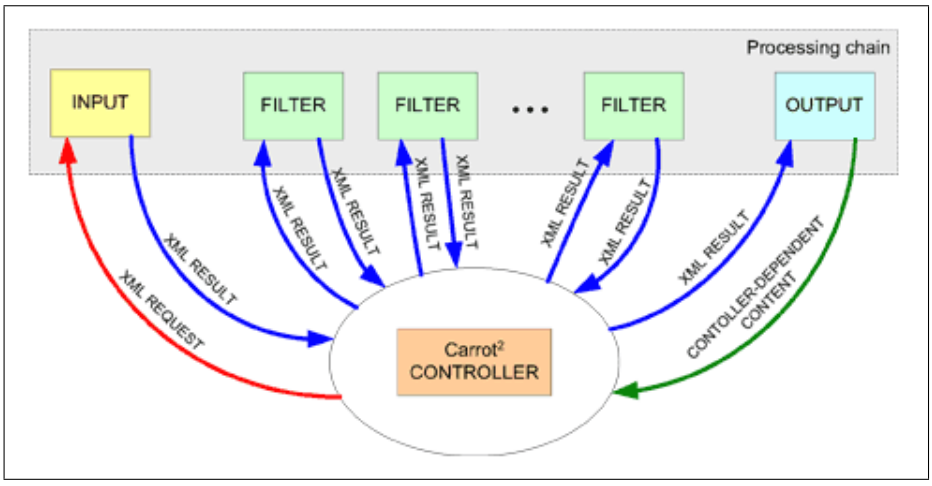
Zij t de term die door defuzzificatie van S bekomen werd als beschrijving voor de cluster C. Aangezien enkel de documenten uit $C_{0.3}$ getoond worden, beschouwen we enkel deze documenten. Zij $l_i, l_{i-1}, \ldots, l_1$ de langste opeenvolging van termen die t bij ten minste 75% van de voorkomens in snippers uit $C_{0.3}$ voorafgaan en zij $r_1, r_2, \ldots, r_j$ de langste opeenvolging van termen die t bij ten minste 75% van de voorkomens in snippers uit $C_{0.3}$ opvolgen. De cluster C wordt dan beschreven als de opeenvolging van termen $l_i, l_{i-1}, \ldots, l_1, t, r_1, r_2, \ldots, r_j$.

Hiërarchisch clusteren

Een hiërarchie van clusters is gemakkelijker te doorbladeren dan een niet-hiërarchische clustering. We kunnen een dergelijke hiërarchie bekomen door het volledige algoritme recursief toe te passen. Voor alle clusters C met $|C_{0.3}| > 25$, passen we het volledige algoritme recursief toe op de snippers uit $C_{0.3}$. Ook de vaagrelaties $N^{\mathcal{D}}$ en $N^{\mathcal{T}}$ worden opnieuw bepaald. Zij S voor de cluster C gedefinieerd als in (6.7) en $\alpha = 0.50 \cdot hgt(S)$. De termen uit S_α worden bij het opstellen van de term- en documentrelaties beschouwd als zoektermen die door de gebruiker werden opgegeven. De lidmaatschap van een snipper d uit $\mathcal{D}$ in een subcluster D van $C_{0.3}$ wordt gedefinieerd als $CD(d) = \min(C(d), D(d))$. Aangezien een clusterhiërarchie met een grote diepte op deze manier een grote complexiteit met zich zou meebrengen, stellen we de maximale diepte van de hiërarchie gelijk aan drie.

6.3.3 Carrot^2

Voor de implementatie van het algoritme werd gebruik gemaakt van het Carrot^2 raamwerk dat door Dawid Weiss werd ontwikkeld aan de Pozna University of Technology. Carrot^2 is ontworpen voor experimenteel onderzoek m.b.t. het bevragen van informatiebronnen (zoals zoekmachines), het manipuleren van de zoekresultaten en de visualisatie ervan [80]. Bij de ontwerpsbeslissingen werd een zo groot mogelijke flexibiliteit nagestreefd, soms ten koste van de efficiëntie. Een duidelijk voorbeeld hiervan is het gebruik van Java als programmeertaal. Een implementatie voor het clusteren van zoekresultaten bestaat uit verschillende functioneel



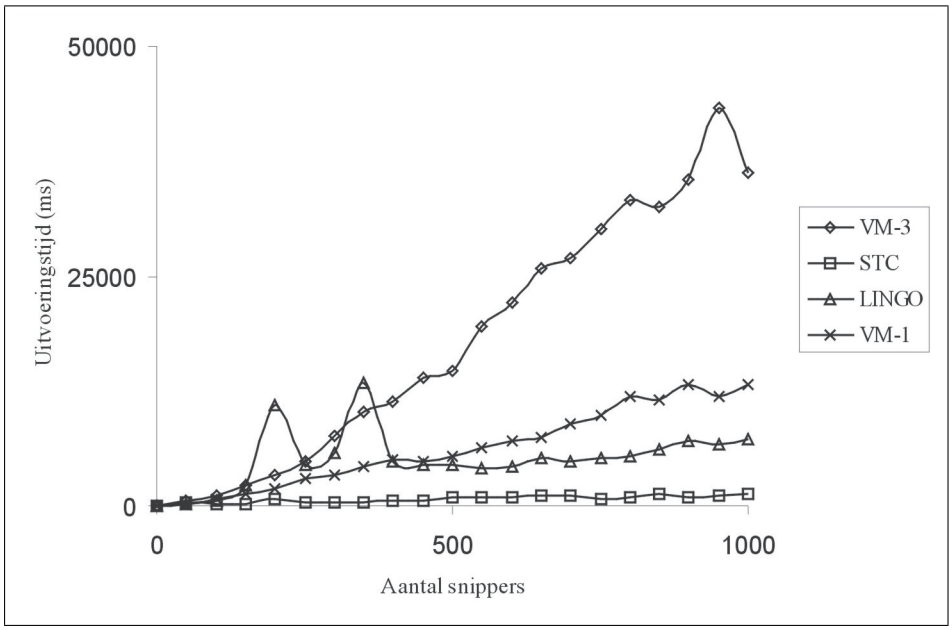
Figuur 6.1: Interactie tussen de Carrot^2 componenten

onafhankelijke deelprocessen: interactie met de zoekmachine, omzetten van termen naar hun grammaticale stamvorm en het verwijderen van stopwoorden, het clusteren van de resultaten en uiteindelijk de presentatie van de clusters. Carrot^2 beschouwt een dergelijk proces als de interactie van een invoercomponent, één of meerdere filtercomponenten en een uitvoercomponent, met een controlecomponent. Dit wordt geïllustreerd in figuur 6.1. Voor onze implementatie hebben we gebruik gemaakt van een invoercomponent die de interactie verzorgt met de zoekmachine Google, van een filtercomponent die de Porter-stemmer implementeert en Engelstalige stopwoorden aanduidt en van de standaard uitvoer- en controlecomponent.

Een nieuwe filtercomponent werd geschreven die het vaagmieralgoritme implementeert.

6.4 Evaluatie

Door de verschillende aanpassingen aan het algoritme kunnen we niet zonder meer onze vaststelling uit hoofdstuk 4 i.v.m. de complexiteit van het algoritme overnemen. Bovendien is het onduidelijk hoeveel tijd het opstellen van de optredende vaagrelaties vergt. Figuur 6.2 toont de uitvoeringstijd voor het vaagmieralgoritme met recursie tot maximaal diepte drie (VM-3), het STC algoritme, LINGO en het vaagmieralgoritme zonder recusieve toepassing (VM-1). LINGO is een algoritme voor het clusteren van snippers dat gebruik maakt van LSI om relaties tussen termen in rekening te brengen [61]. De figuur illustreert duidelijk



Figuur 6.2: Uitvoeringstijd voor de zoekbewerking “fuzzy logic”

de superioriteit van STC wat de uitvoeringssnelheid betreft. Het vaagmieralgoritme met recursieve toepassing is duidelijk heel wat trager dan zowel LINGO en STC. Aangezien LINGO en STC een vlakke partitionering genereren, is het zinvol om bij het vergelijken enkel het eerste niveau van de recursie te beschouwen. Hiervoor zien we dat het vaagmieralgoritme inzake uitvoeringssnelheid vergelijkbaar is met LINGO. In een geoptimaliseerde implementatie zou het genereren van subclusters kunnen uitgesteld worden tot het moment deze opgevraagd worden. Een dergelijke implementatie zou resulteren in een globale snelheidswinst aangezien typisch maar een klein deel van de subclusters zullen opgevraagd worden. Bovendien kan het berekenen van de subclusters redelijk snel gebeuren. Deze aanpak kan echter niet gerealiseerd worden binnen het huidige Carrot^2 raamwerk.

Een kwalitatieve beoordeling van een algoritme om zoekresultaten te clusteren is inherent subjectief. Documentenverzamelingen met een gekende structuur, zoals de TREC-documentenverzamelingen, zijn voor een dergelijke evaluatie niet bruikbaar [81]. De imple-

mentatie van het algoritme is beschikbaar op de publieke demo van het Carrot^2-project[8]. In bijlage A worden de clusters getoond die bekomen werden met het vaagmieralgoritme, LINGO, STC en Vivisimo voor een vijftal zoekopdrachten. Zowel het vaagmieralgoritme als het algoritme dat door Vivisimo gebruikt wordt, zijn niet deterministisch in die zin dat verschillende uitvoeringen van het algoritme verschillende clusters kunnen opleveren. De getoonde resultaten zijn bekomen voor een willekeurige uitvoering. Voor alle zoekopdrachten werden de eerste 200 snippers geclusterd. Dit correspondeert met het standaard aantal snippers dat door Vivisimo geclusterd wordt. Aangezien Vivisimo geen gebruik maakt van de zoekresultaten van Google, kunnen kwalitatieve verschillen tussen de resultaten van Vivisimo en de andere algoritmen hierdoor beïnvloed zijn.

In het algemeen kunnen we stellen dat het aantal clusters bij het vaagmieralgoritme significant kleiner is dan bij de andere algoritmen. Een belangrijk nadeel van een groot aantal clusters is dat de belangrijkste clusters samen maar een klein deel van de volledige snipperverzameling omvatten. De vele kleine clusters behandelen bovendien typisch verschillende malen hetzelfde onderwerp. De clusters weerspiegelen m.a.w. niet de structuur van de snipperverzameling. Vermoedelijk is dit een gevolg van het feit dat voor de onderliggende algoritmen het aantal clusters vooraf moet gekend zijn, waardoor gebruik gemaakt moet worden van onbetrouwbare heuristieken. Anderzijds zijn de labels van te algemene clusters niet altijd voldoende informatief. Dit wordt geïllustreerd door de resultaten van de zoekopdracht "Nature Inspired Algorithms". Het vaagmieralgoritme vindt één hoofdcluster die beschreven wordt m.b.v. het label "genetic algorithms". De reden hiervoor is dat de meeste snippers in deze cluster over genetische algoritmen handelen. Nochtans komen bijvoorbeeld ook snippers voor i.v.m. gesimuleerde tempering (simulated annealing) en miergebaseerde algoritmen.

[8] http://carrot.cs.put.poznan.pl

Hoofdstuk 7

Samenvatting

Biologisch geïnspireerde algoritmen pogen de oplossing van een zeker complex probleem te bekomen door gedrag dat in de natuur werd geobserveerd na te bootsen. In hoofdstuk 1 hebben we enkele aspecten van het gedrag van mieren besproken. Hieruit bleek hoe het complexe gedrag van een kolonie mieren steeds ontstaat als het samengestelde effect van eenvoudige gedragingen van de verschillende individuen. In het bijzonder hebben we het model van Deneubourg et al. besproken voor het clusteren van dode soortgenoten, zoals dit bij verschillende miersoorten werd geobserveerd. Dit model leidde Lumer en Faieta er in 1994 toe om een clusteringsalgoritme op te stellen dat hierop gebaseerd is.

Dit algoritme en de belangrijkste varianten ervan, werden besproken in hoofdstuk 2. Tevens werden de belangrijkste tekortkomingen van deze algoritmen geïdentificeerd. Ondanks hun vele aantrekkelijke eigenschappen, slagen deze algoritmen er onvoldoende in om op gepaste wijze een abstractie te maken van het probleem. Een eerste stap in de goede richting werd gegeven door Monmarché, die voorstelde om de roosterrepresentatie die afkomstig is van het model van Deneubourg et al., voor het clusteringsprobleem achterwege te laten. Hij stelde een algoritme voor waarin de (artificiële) mieren zowel individuele objecten als volledige hopen van objecten kunnen verplaatsen. Om dit te realiseren stelde hij voor om in twee fasen te werken. In de eerste fase verplaatsen mieren slechts individuele objecten, in de tweede fase volledige hopen. Door de ruisgevoeligheid van de tweede fase, is het toepassen van het k-gemiddelde algoritme na de eerste fase noodzakelijk; omdat geen individuele objecten kunnen worden weggenomen in de tweede fase is het toepassen van het k-gemiddelde algoritme bovendien ook noodzakelijk na de tweede fase.

In hoofdstuk 4 hebben we, geïnspireerd door het werk van Monmarché, een nieuw clusteringsalgoritme geïntroduceerd. We hebben getoond hoe we, door gebruik te maken van een model voor taakverdeling bij sociale insecten van Bonabeau et al., zowel het gebruik van meerdere fasen als de hybridisatie met k-gemiddelde kunnen vermijden. Het bepalen van de optredende stimuliwaarden in dit model, werd mogelijk gemaakt door de flexibiliteit die een beschrijving a.d.h.v. vaagregels met zich meebrengt. De experimentele resultaten suggereren dat het algoritme een robuuste oplossing met een hoge efficiëntie kan bekomen, waarbij de hoofdclusters van de gebruikte gegevensverzameling ontdekt worden zonder gebruik te maken van a priori informatie. Bovendien werd bij de bestudeerde reële gegevensverzamelingen een significante verbetering tegenover het algoritme van Monmarché m.b.t. de classificatiefout vastgesteld.

Het belangrijkste voordeel van miergebaseerde clusteringsalgoritmen in het algemeen, is

dat het aantal clusters in de te clusteren gegevensverzameling niet op voorhand hoeft gekend te zijn. In de context van het internet zijn tal van voorbeelden te bedenken waarbij men a priori volledig in het duister tast omtrent de structuur van de te clusteren gegevensverzamelingen. In deze scriptie hebben we ons verder toegelegd op het clusteren van de zoekresultaten, zoals die door een zoekmachine teruggegeven worden. Een belangrijk aspect hierbij is dat het clusteren moet gebeuren op basis van een kort uitreksel van elke pagina. In hoofdstuk 5 hebben we een aantal mogelijkheden bestudeerd om documenten te vergelijken. In het bijzonder werden similariteit en de relatie "specifieker dan" besproken. Omdat we slechts korte uitreksels van elk document gebruiken, is het noodzakelijk om bij het vergelijken van documenten verder te gaan dan het louter vergelijken van de optredende termen in deze documenten. Door gebruik te maken van formele vaagconcept-analyse hebben we op een natuurlijke manier vaagrelaties afgeleid die de similariteit en de relatie "specifieker dan" voor documenten modelleren. Relaties tussen termen worden expliciet in rekening gebracht door gebruik te maken van een hybridisatie met formele ruwconcept-analyse. We hebben tevens aangegeven waar de klassieke aanpak, die gebaseerd is op dimensiereductiemethoden uit de lineaire algebra, tekort schiet.

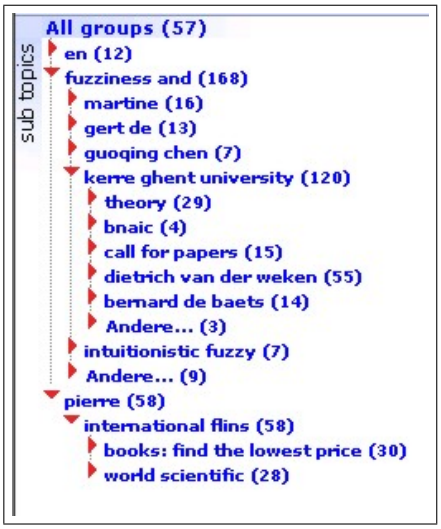
In hoofdstuk 6 ten slotte, werden enkele aanpassingen aan het vaagmieralgoritme uit hoofdstuk 4 voorgesteld om te komen tot een algoritme voor het clusteren van zoekresultaten. In tegenstelling tot de klassieke clusteringsalgoritmen voor zoekresultaten, wordt een overzichtelijke hiërarchie teruggegeven met de hoofdclusters uit de onderliggende gegevensverzameling. Het geringe aantal clusters in het resultaat is een gevolg van drie verschillende aspecten:

- Het vaagmieralgoritme zoekt steeds de hoofdclusters in een gegevensverzameling.
- Door gebruik te maken van de bovenbenadering van de met de gegevensverzameling corresponderende formele vaagcontext, worden veel meer paren van documenten in een strikt positieve mate verwant beschouwd.
- Het gebruik van de relatie "specifieker dan" in plaats van een similariteitsmaat, zorgt eveneens voor een stijging van de mate van verwantschap tussen de verschillende paren van documenten.

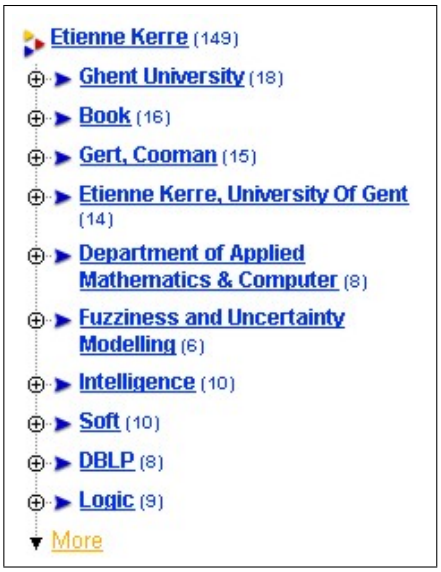
Of, en in welke situaties, een overzicht met hoofdclusters te prefereren is boven een overzicht met meer gedetailleerde clusters, blijft een open vraag en is ongetwijfeld in sterke mate gebruikersafhankelijk.

Bijlage A

Voorbeelden van het clusteren van snippers

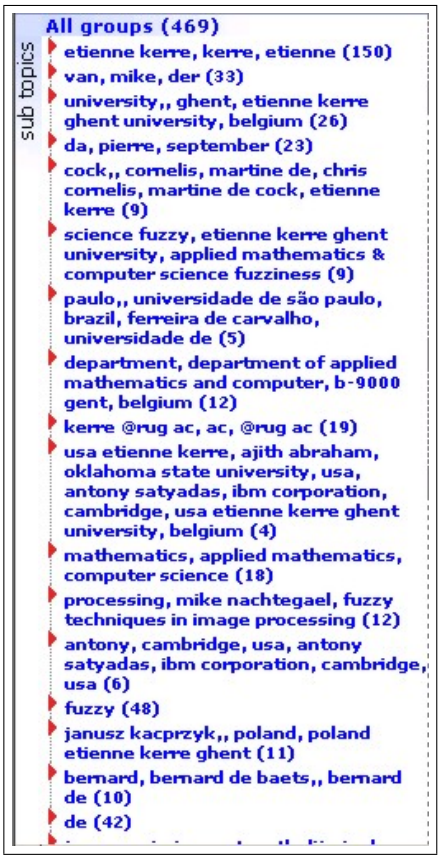


(Vaagmieren)



(Vivisimo)

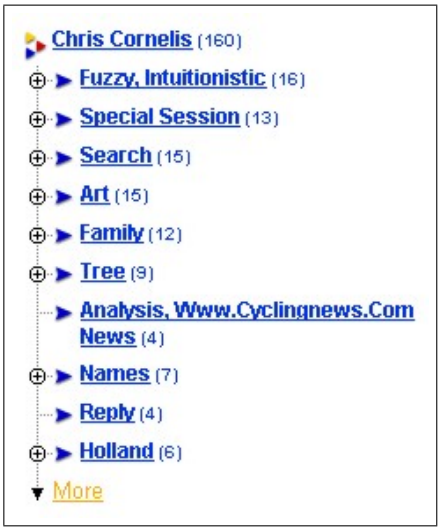
(LINGO)



(STC)

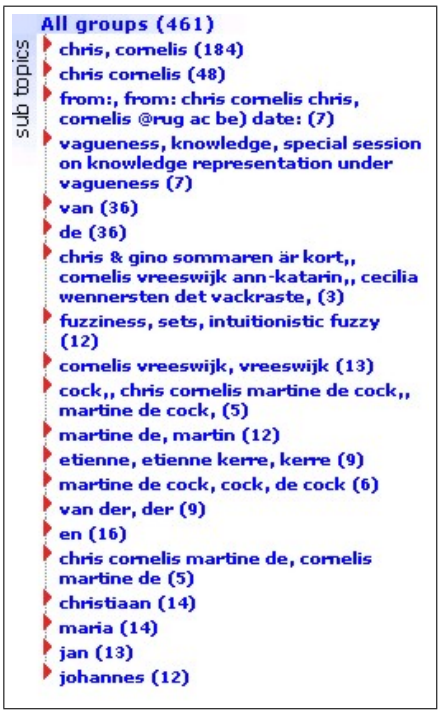
Figuur A.1: Resultaten voor “Etienne Kerre”, 200 snippers

(Vaagmieren)



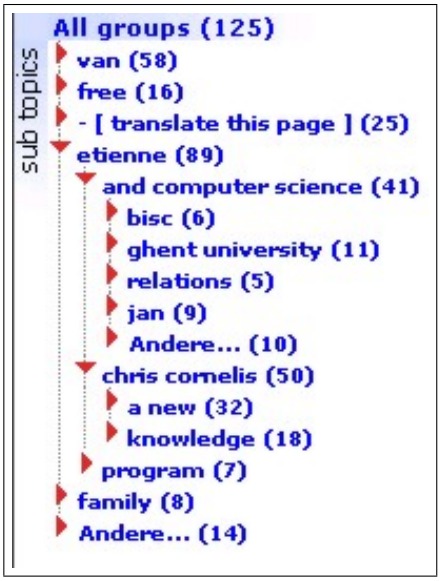
(Vivisimo)

(LINGO)

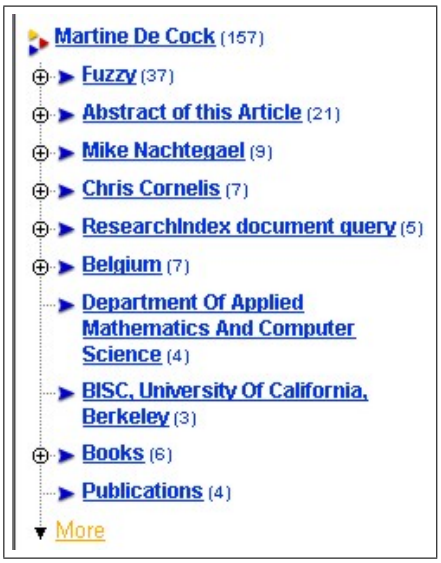


(STC)

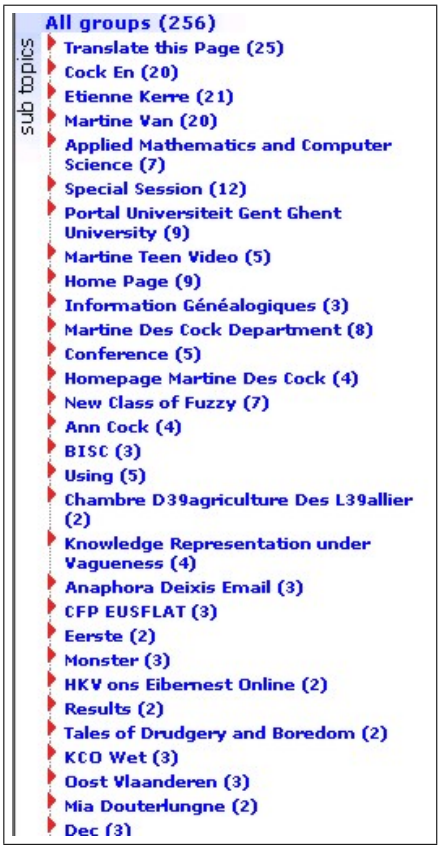
Figuur A.2: Resultaten voor “Chris Cornelis”, 200 snippers



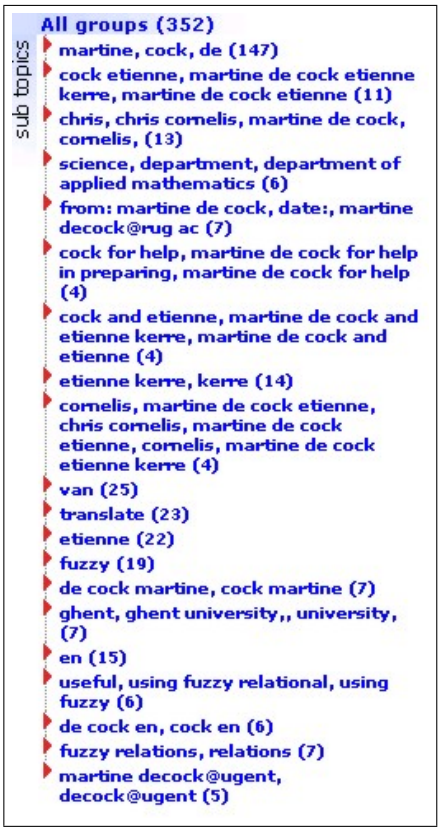
(Vaagmieren)



(Vivisimo)

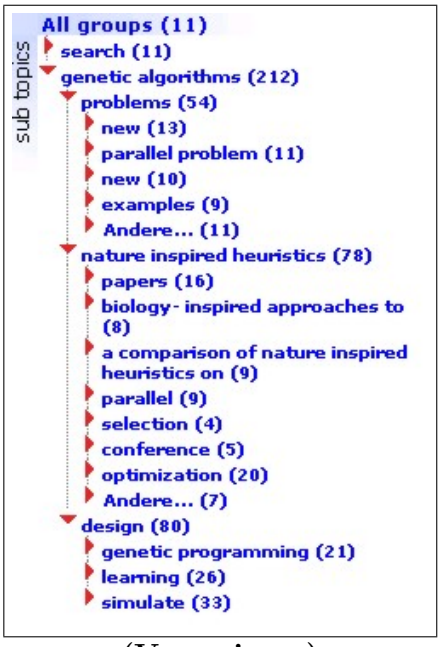


(LINGO)

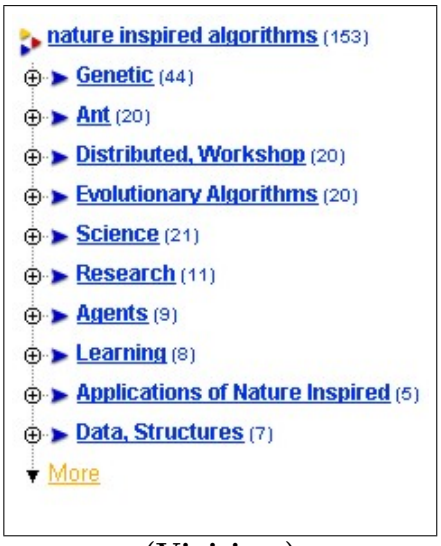


(STC)

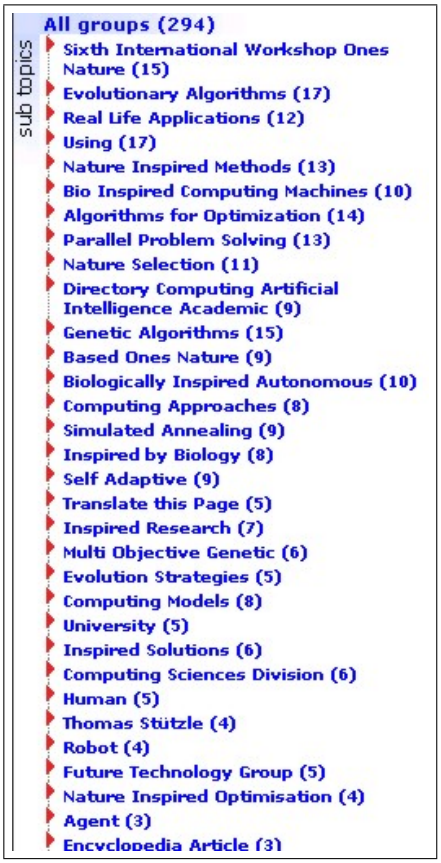
Figuur A.3: Resultaten voor “Martine De Cock”, 200 snippers



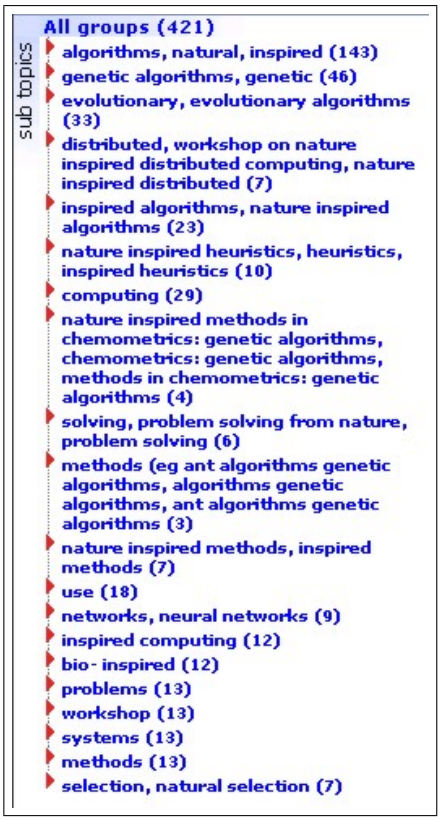
(Vaagmieren)



(Vivisimo)



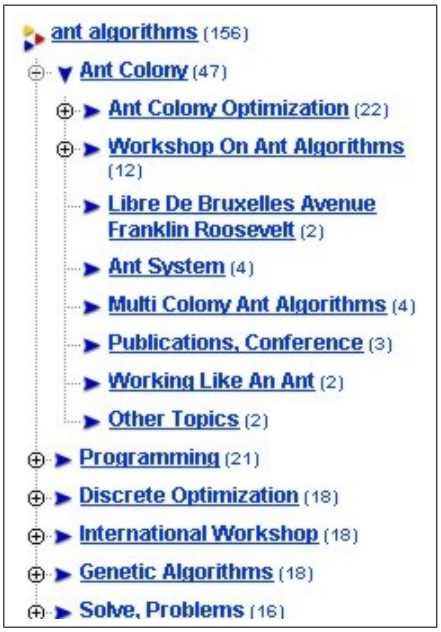
(LINGO)



(STC)

Figuur A.4: Resultaten voor “Nature Inspired Algorithms”, 200 snippers

(Vaagmieren)



(Vivisimo)

(LINGO)

(STC)

Figuur A.5: Resultaten voor “Ant Algorithms”, 200 snippers

Bijlage B

Enkele klassieke clusteringsalgoritmen

B.1 Partitiegebaseerde algoritmen

Partitiegebaseerde clusteringsalgoritmen proberen een verzameling $X = \{x_1, x_2, \ldots, x_n\}$ van objecten voor te stellen als een verzameling van clusters $\{C_1, C_2, \ldots, C_k\}$ waarbij objecten die tot dezelfde cluster behoren meer gelijkaardig zijn dan objecten uit verschillende clusters. Een cluster wordt hierbij meestal gedefinieerd als een (scherpe) deelverzameling van X. Voor clusters C_i en C_j, met i en j in $\{1, 2, \ldots, n\}$, wordt bovendien meestal ondersteld dat $C_i \cap C_j = \emptyset$. Bij vaagclusteringsalgoritmen wordt een cluster gedefinieerd als een vaagverzameling in X en kunnen objecten tot verschillende clusters, in een zekere mate, behoren. We bespreken nu enkele gekende partitiegebaseerde clusteringsalgoritmen.

k-gemiddelde (k-means) Als de objecten gekenmerkt worden door numerieke attributen, m.a.w. wanneer we de objecten kunnen interpreteren als vectoren, kan gebruik gemaakt worden van het k-gemiddelde algoritme. Zij $x_i = (a_{i1}, a_{i2}, \ldots, a_{im})$ voor i in $\{1, 2, \ldots, n\}$. We veronderstellen dat het aantal clusters op voorhand gekend is en dat een initiële definitie van de clusters $C_1, C_2, \ldots, C_k$ beschikbaar is. Deze initiële clusterdefinitie kan bijvoorbeeld willekeurig bepaald worden. Elke cluster C_i ($i \in \{1, 2, \ldots, k\}$) wordt gekenmerkt door zijn centroïde $v_i = (\overline{a_1^i}, \overline{a_2^i}, \ldots, \overline{a_m^i})$. Hierbij is $\overline{a_j^i} = \frac{1}{|C_i|} \sum_{x_k \in C_i} a_{kj}$ voor j in $\{1, 2, \ldots, m\}$. Veronderstellen we bovendien dat d de dissimilariteit tussen de objecten en centroïden uitgedrukt; we stellen d bijvoorbeeld gelijk aan de Euclidische afstand. Het k-gemiddelde algoritme verloopt als volgt.

1. Bereken de de centroïden $v_1, v_2, \ldots, v_k$ van de huidige partitie $C_1, C_2, \ldots, C_n$
2. Definieer C_i voor i in $\{1, 2, \ldots, k\}$ als

   $$C_i = \{x | x \in X \wedge d(x, v_i) = \min_{j \in \{1,2,\ldots,k\}} d(x, v_j)\}$$

   Wanneer $\min_{j \in \{1,2,\ldots,k\}} d(x, v_j) = d(x, v_{i_1}) = d(x, v_{i_2})$ voor zekere $i_1 \neq i_2$, wordt willekeurig bepaald of x toegevoegd wordt aan C_{i_1} of aan C_{i_2}. In geen geval wordt x aan meer dan één cluster toegekend.
3. Als een zeker stopcriterium nog niet voldaan is: keer terug naar stap 2.

Partitioneren rond medoïden (pam, partitioning around medoids) Wanneer objecten niet als vectoren kunnen beschouwd worden, kunnen we de centroïden uit het k-gemiddelde algoritme niet bepalen. Een cluster kan dan voorgesteld worden door het meest representatieve object van die cluster. Dit meest representatieve object wordt een medoïde genoemd. Zij k opnieuw het gewenste aantal clusters. De medoïden $m_1, m_2, \ldots, m_k$ worden als volgt iteratief bepaald [79].

1. Kies m_1 als het object m in X waarvoor
$$\sum_{i=1}^{n} d(x_i, m)$$
minimaal is. Voor $j = 2, 3, \ldots, k$ wordt m_j in X gekozen zodat
$$\sum_{i=1}^{n} \min_{l=1}^{j} d(x_i, m_l)$$
minimaal is.

2. Beschouw alle paren (m, x) met $m \in \{m_1, m_2, \ldots, m_k\}$ en $x \in X \setminus \{m_1, m_2, \ldots, m_k\}$. Voor elk dergelijk paar bepalen we of de objectieffunctie
$$\sum_{i=1}^{n} \min_{l=1}^{k} d(x_i, m_l)$$
verminderd wordt door de medoïde m te vervangen door het object x. Als dit het geval is voor bepaalde paren, noemen we het paar dat de objectieffunctie het meest vermindert (m^*, x^*) en wordt de medoïde m^* vervangen door het object x^*. Alle andere medoïden blijven dezelfde. Als geen dergelijke paren bestaan, eindigt het algoritme.

3. Herhaal stap 2 tot convergentie optreedt.

Wanneer de medoïden bepaald zijn, wordt elk object toegekend aan de cluster waarvoor de dissimilariteit met de corresponderende medoïde minimaal is. Wanneer deze dissimilariteit minimaal is voor verschillende medoïden, wordt een willekeurige keuze gemaakt.

Vaag c-gemiddelde (fuzzy c-means) Het vaag c-gemiddelde algoritme is een variant van het k-gemiddelde algoritme, waarbij een object in een zekere mate kan behoren tot verschillende clusters. Een cluster wordt bijgevolg voorgesteld als een vaagverzameling in X. Zij $\{v_1, v_2, \ldots, v_k\}$ de centroïden van de clusters. Noteren we de lidmaatschapsgraad van het object x_j in de cluster C_i als $C_i(x_j)$ ($i \in \{1, 2, \ldots, k\}$, $j \in \{1, 2, \ldots, n\}$). Veronderstellen we dat een initiële waarde voor deze centroïden gekend is; deze kunnen bijvoorbeeld willekeurig gekozen worden. Bovendien onderstellen we nog steeds dat d de dissimilariteit tussen objecten en centroïden uitdrukt. De lidmaatschapsgraden worden dan bepaald d.m.v. de volgende iteratieve procedure [13].

1. Voor $i \in \{1, 2, \ldots, k\}$ en $j \in \{1, 2, \ldots, n\}$ definiëren we $C_i(x_j)$ als:
$$C_i(x_j) = \begin{cases} \dfrac{1}{\displaystyle\sum_{l=1}^{k} \left(\dfrac{d(v_i, x_j)}{d(v_l, x_j)} \right)^{\frac{2}{M-1}}} & \text{Als } d(v_l, x_j), \text{ voor alle } l \text{ in } \{1, 2, \ldots, k\} \\ 1 & \text{Als } d(v_i, x_j) = 0 \\ 0 & \text{Anders} \end{cases}$$

2. Het clustercentrum v_i, met i in $\{1, 2, \ldots, k\}$, wordt gedefinieerd als

$$v_i = \frac{\sum_{l=1}^{n} (C_i(x_l))^M x_l}{\sum_{l=1}^{n} (C_i(x_l))^M}$$

   waarbij de objecten worden opgevat als vectoren.

3. Keer terug naar stap 1 als nog geen convergentie bereikt is.

Hierbij is M een reële parameter in $]1, +\infty[$.

B.2 Hiërarchische algoritmen

Hiërarchische clusteringsalgoritmen stellen een gegevensverzameling voor als een hiërarchie van clusters. Elke cluster kunnen we hierbij conceptueel nog steeds als een (scherpe) verzameling voorstellen. Hiërarchische clusteringsalgoritmen kunnen onderverdeeld worden in twee groepen: agglomeratieve en splitsende (divisive) algoritmen.

Agglomeratieve algoritmen beschouwen initieel elk object van de gegevensverzameling als een afzonderlijke cluster. In elke stap worden de twee clusters die het meest gelijkend zijn, samengenomen, tot uiteindelijk een cluster bekomen wordt die de volledige gegevensverzameling omvat. De similariteit tussen clusters kan hierbij op verschillende manieren gedefinieerd worden. Zij C en C' twee clusters, de similariteit $s^*(C, C')$ tussen deze clusters kan dan bijvoorbeeld gedefinieerd worden als [79][57]

Complete link (complete-link)

$$s^*(C, C') = \min_{x \in C} \min_{y \in C'} s(x, y)$$

Enkelvoudige link (single-link)

$$s^*(C, C') = \max_{x \in C} \max_{y \in C'} s(x, y)$$

Gemiddelde link (average-link)

$$s^*(C, C') = \frac{1}{|C| \cdot |C'|} \sum_{x \in C} \sum_{y \in C'} s(x, y)$$

Hierbij is s een zekere similariteitsmaat voor individuele objecten. Een naïeve implementatie van deze drie alternatieven vereist een uitvoeringstijd die voor een gegevensverzameling van grootte n evenredig is met n^3. Voor enkelvoudige link is een implementatie bekend met een uitvoeringstijd die evenredig is met n^2; voor complete link is een uitvoeringstijd evenredig met $n^2 \log n$ mogelijk [57].

Splitsende algoritmen beschouwen de volledige gegevensverzameling initieel als één cluster. Vervolgens wordt in elke stap een cluster geselecteerd die wordt opgesplitst in twee afzonderlijke clusters. Dit wordt, in principe, herhaald tot elke cluster bestaat uit een individueel object. Verschillende algoritmen onderscheiden zich onder meer door de keuze van de te splitsen cluster in elke stap en de manier waarop de splitsing uitgevoerd wordt.

Zowel agglomeratieve als splitsende algoritmen genereren dus een boomstructuur met een bijzondere structuur: elke interne knoop heeft exact twee kinderen. Een dergelijke boomstructuur wordt meestal een dendogram genoemd. Snoeitechnieken kunnen dan toegepast worden om het bekomen dendogram om te vormen tot een meer leesbare boom, waarbij interne knopen typisch veel meer dan twee kinderen zullen hebben. Een belangrijk voordeel van hiërarchische clusteringsalgoritmen is dat het aantal clusters niet expliciet moet opgegeven worden. De gebruikte snoeitechnieken zijn echter wel afhankelijk van een drempelwaarde voor een zeker coherentiecriterium voor de clusters. Een belangrijk nadeel van (agglomeratieve) hiërarchische clusteringsalgoritmen is hun hoge computationele kost

Bijlage C

Implementatie-aspecten

Het vaagmieralgoritme uit hoofdstuk 4 en de aanpassingen voor het clusteren van zoekresultaten uit hoofdstuk 6, werden geïmplementeerd m.b.v. de programmeertaal Java. We bespreken in deze laatste bijlage heel beknopt de broncode van beide implementaties. Voor meer gedetailleerde informatie verwijzen we naar het commentaar in de broncode zelf. De broncode van beide implementaties is beschikbaar op de cd die bij deze scriptie werd geleverd; de broncode van de Carrot^2-component is eveneens beschikbaar op de webpagina van het Carrot^2-project[1].

C.1 Clusteren van objecten

De hoofdklasse van de implementatie van het vaagmieralgoritme uit hoofdstuk 4, is de klasse `LijstMain`. De klasse `LijstModel` zorgt voor de wisselwerking tussen het clusteringsproces en de grafische representatie. Het eigenlijke clusteringsproces ten slotte, wordt afgehandeld in de klasse `LijstMier`. Deze klasse bevat niet alleen de coördinatie van het algoritme en de gebruikte vaagregels, maar definieert ook hoe de inferentie met de vaagregels moet gebeuren. Dit wordt geïllustreerd in het volgend codefragment.

```
public static void infer(FuzzyNumber f1,int w1,FuzzyNumber f2, int w2,
                         FuzzyNumber f3,FuzzyNumber res){
    if(f1.lidmaatschap(w1)>0 && f2.lidmaatschap(w2)>0){
        FuzzyNumber temp=new FuzzyNumber(f3,Math.min(f1.lidmaatschap(w1),
                                                     f2.lidmaatschap(w2)));
        res.max(temp);
    }
}
```

Het codefragment implementeert inferentie voor de vaagregel

$$\text{Als } X_1 \text{ is } f_1 \text{ en } X_2 \text{ is } f_2 \text{ dan } res \text{ is } f_3$$

waarbij `w1` en `w2` de scherpe waarneming van X_1 en X_2 voorstellen. Wanneer verschillende vaagregels gebruikt worden, wordt het maximum als aggregatie-operator gebruikt. De vaagregels worden bovendien gemodelleerd m.b.v. het minimum. In de klasse `FuzzyNumber` werden

[1] http://sourceforge.net/projects/carrot2

de nodige operatoren geïmplementeerd om deze inferentie te realiseren. De te clusteren objecten worden voorgesteld door de klasse `KleurDocument` die naast de numerieke attributen van het object, ook een kleur bevat voor de grafische weergave. De klasse `Hoop` stelt een hoop van objecten voor en definieert onder meer het centrum van een hoop en de berekening van de verschillende karakteristieken van een hoop, zoals de gemiddelde en minimale similariteit met het centrum.

C.2 Clusteren van zoekresultaten

Het vaagmieralgoritme voor het clusteren van zoekresultaten werd, zoals we reeds in hoofdstuk 6 stelden, geïmplementeerd als filtercomponent van het Carrot[2] raamwerk. Het volstaat hiertoe een uitbreiding te definiëren van de abstracte klasse `FilterRequestProcessor` die de methode `processFilterRequest` implementeert. Deze methode krijgt als input o.a. een XML[2]-boom die de snippers van de zoekbewerking bevat. Elke filtercomponent voegt specifieke knopen aan deze XML-boom toe; zo zal een stemmer bijvoorbeeld knopen toevoegen die voor de termen die voorkomen in de snippers, de vertaling zal bevatten naar de corresponderende grammaticale stamvorm. Merk op dat het wijzigen van deze XML-boom de enige manier is waarop informatie tussen verschillende componenten kan uitgewisseld worden. Een filtercomponent die snippers wenst te clusteren, zal dus knopen moeten toevoegen waarin deze clusters gedefinieerd worden. Deze informatie kan vervolgens door de uitvoercomponent gebruikt worden om de gevonden clusters weer te geven.

De vereiste uitbreiding van de abstracte klasse `FilterRequestProcessor` wordt gerealiseerd door de klasse `FuzzyAnts` die m.a.w. de hoofdklasse van de filtercomponent vormt. De klasse `DocumentClustering` coördineert het eigenlijke clusteringsproces en bevat onder meer de code die de recursieve toepassing van het algoritme realiseert, de labels van de clusters bepaalt en de gewichten van de snippers in elke cluster definieert. Het eigenlijke vaagmieralgoritme wordt gecoördineerd door de klasse `ListBordModel`. In de klasse `SnippetParser` worden de vaagrelaties die gebruikt worden door het algoritme opgesteld. Een efficiënte implementatie hiervan, die gebruik maakt van het ijle karakter van de matrices die we kunnen associëren met deze vaagrelaties, is van primordiaal belang om een aanvaardbare uitvoeringstijd te bekomen. Hiertoe wordt gebruik gemaakt van de standaardklasse `HashMap` uit het `java.util` pakket (package). Dit is een efficiënte implementatie van het abstracte datatype `Map`, die gebruik maakt van hashtabellen. Een `Map` kan conceptueel beschouwd worden als een verzameling van koppels van (willekeurige) objecten en stelt bijgevolg een functie voor. We illustreren dit a.d.h.v. de methode `narrowTerms` die de voorverzameling $N^{\mathcal{T}}t$ bepaalt voor een term t uit $\mathcal{T}$.

```
private HashMap narrowTerms(int t){
        if (narrowTerms == null)
            narrowTerms = new HashMap();
        if (narrowTerms.keySet().contains(new Integer(t)))
            return (HashMap) narrowTerms.get(new Integer(t));

        HashSet universe = new HashSet();
        HashMap result = new HashMap();
```

[2]eXtended Markup Language

```
        for (Iterator it = docWeights[t].keySet().iterator(); it.hasNext();){
            int doc = ((Integer) it.next()).intValue();
            universe.addAll(termWeights[doc].keySet());
        }

        for (Iterator it = universe.iterator(); it.hasNext();){
            Integer termIndex = (Integer) it.next();
            if (t == termIndex.intValue())
                result.put(termIndex, new Double(1));
            else{
                double specValue = narrowTerm(termIndex.intValue(), t);
                if (specValue > 0.30)
                    result.put(termIndex, new Double(specValue));
            }
        }

        narrowTerms.put(new Integer(t), result);
        return result;
    }
```

Het resultaat van deze methode, wordt toegevoegd aan de `HashMap narrowTerms`, zodat de voorverzameling $N^{\mathcal{T}}t$ voor elke t in $\mathcal{T}$ ten hoogste één maal moet bepaald worden. In de verzameling `universe` worden alle termen opgenomen die in ten minste één snipper samen voorkomen met t. Bijgevolg wordt $N_1^{\mathcal{T}}(t', t)$ op deze manier slechts geëvalueerd voor termen t' die aanleiding geven tot een strikt positieve waarde. Ten slotte worden enkel de termen t' waarvoor $N_1^{\mathcal{T}}(t', t) > 0.3$ behouden.

Bibliografie

[1] P. Albuquerque, A. Dupuis. A Parallel Cellular Ant Colony Algorithm for Clustering and Sorting. *Proceedings of the 5th International Conference on Cellular Automata for Research and Industry.* pp. 220-230, 2002.

[2] H. Azzag, N. Monmarché, M. Slimane, C. Guinot, G. Venturini. Algorithme AntTree: Classification non supervisée par des fourmis artificielles. *Revue des nouvelles technologies et l'information.* pp. 75-86, 2003.

[3] R. Baeza-Yates. Web Usage Mining in Search Engines. Aanvaard voor publicatie in "Web Mining: Applications and Techniques", beschikbaar op "http://db.uwaterloo.ca/seminars/notes/ricardo.pdf", 2004.

[4] R. Beckers, O. E. Holland, and J. L. Deneubourg. From Local Actions To Global Tasks: Stigmergy And Collective Robotics. *Artificial Life IV: Proceedings of the 4th International Workshop on the Synthesis and Simulation of Living Systems.* pp. 181-189, 1994.

[5] M. Beekman, D.J.T. Sumpter, F.L.W. Ratnieks. Phase transitions between disordered and ordered foraging in Pharao's ants. *Proceedings of the National Academy of Sciences of the Unites States of America.* Vol. 98 no. 17, pp. 9703-9706, 2001.

[6] R. Bělohlávek. Similarity Relations in Concept Lattices. *Journal of Logic Computations.* Vol. 10 No. 6, pp. 823-845, 2000.

[7] M.W. Berry, S.T.Dumais, G.W. O'Brian. Using Linear Algebra for Intelligent Information Retrieval. *SIAM review.* Vol. 37, No. 4, pp. 573-595, 1994.

[8] W. Bin, Z. Yi, L. Shaohui, S. Zhongzhi. CSIM: A Document Clustering Algorithm Based On Swarm Intelligence. *Proceedings of the 2002 Congress on Evolutionary Computation Conference.* Vol. 1, pp. 477-482, 2002.

[9] C.L. Blake, C.J. Merz. UCI Repository of machine learning databases. Beschikbaar op http://www.ics.uci.edu/ mlearn/MLRepository.html, University of California, 1998.

[10] U. Bodenhofer. A New Approach to Fuzzy Ordenings. *Tatra Mt. Math. Publ.* Vol. 16 Part I, pp. 21-29, 1999.

[11] E. Bonabeau. From Classical Models of Morphogenesis to Agent-Based Models of Pattern Formation. *Artificial Life III: Proceedings of the 3th International Workshop on the Synthesis and Simulation of Living Systems.* no. 3, pp. 191-211, 1997.

[12] E. Bonabeau, A. Sobkowski, G. Theraulaz, J.L. Deneubourg. Adaptive Task Allocation Inspired by a Model of Division of Labor in Social Insects. Working Paper 98-01-004, Santa Fe Institute, beschikbaar op "http://ideas.repec.org/p/wop/safiwp/98-01-004.html", 1998.

[13] T.W. Cheng, D.B. Goldgof, L.O. Hall. Fast Fuzzy Clustering. *Fuzzy Sets and Systems.* Vol. 93, pp. 49-56, 1998.

[14] D.R. Chialvo, M.M. Millonas. How Swarms Build Cognitive Maps. *The Biology and Technology of Intelligent Autonomous Agents: Proceedings of the NATO Advanced Study Institute on The Biology and Technology of Intelligent Autonomous Agents.* Vol. 144, pp. 439-450, NATO ASI Series, Springer, 1995.

[15] C. Cornelis. *Benaderend redeneren in de vaagverzamelingenleer.* Licentiaatsscriptie, Universiteit Gent, 2000.

[16] C. Cornelis. *Tweezijdigheid in de Representatie en Verwerking van Imprecieze Informatie.* Doctoraatsproefschrift, Universiteit Gent, 2004.

[17] D.R. Cutting, D.R. Karger, J.O. Pedersen, J.W. Tukey. Scatter/Gather: A Cluster-based Approach to Browsing Large Document Collections. *Proceedings of the 15th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval.* pp. 318-329, 1992.

[18] B. De Baets. *Oplossen van vaagrelationele vergelijkingen: een ordetheoretische benadering.* Doctoraatsproefschrift, Universiteit Gent, 1995.

[19] B. De Baets, H. De Meyer, The Frank t-norm family in fuzzy similarity measurement. *Proceedings of the Second EUSFLAT Conference.* pp. 249-252, 2001.

[20] M. De Cock. *Een grondige studie van linguïstische wijzigers in de vaagverzamelingenleer.* Doctoraatsproefschrift, Universiteit Gent, 2002.

[21] M. De Cock. *Cursusnota's bij het opleidingsonderdeel "Gastcolleges over actuele aspecten in de informatica: trends in soft computing".* Universiteit Gent, 2004.

[22] M. De Cock, C. Cornelis, E. E. Kerre. Elicitation of Fuzzy Association Rules from Positive and Negative Examples. Aanvaard voor publicatie in Fuzzy Sets and Systems.

[23] J.L. Deneubourg, S. Goss, N. Franks, A. Sendova-Franks, C. Detrain, L. Chrétien. The dynamics of collective sorting robot-like ants and ant-like robots. *From Animal to Animats: Proceedings of the First International Conference on Simulation of Adaptive Behaviour.* pp. 356-363, 1990.

[24] I. Dhillon, J. Kogan, C. Nicholas. Feature Selection and Document Clustering. *Survey of Text Mining. Clustering, Classification and Retrieval.* Springer-Verslag New York, pp. 73-100, 2003.

[25] G. Di Caro, M. Dorigo. Mobile Agents for Adaptive Routing. *Proceedings of the 31st Hawaii International Conference on Systems.* pp. 74-83, 1998.

[26] M. Dorigo, G. Di Caro, L. M. Gambardella. Ant Algorithms for Discrete Optimization. *Artificial Life.* Vol. 5 No. 3, pp. 137-172, 1999.

[27] M. Dorigo, L. M. Gambardella. Ant Colonies for the Traveling Salesman Problem. *BioSystems.* Vol. 43, pp. 73-81, 1997.

[28] M. Dorigo, V. Maniezzo, A. Colorni. The Ant System: An Autocatalytic Optimizing Process. Technical Report No. 91-016 Revised, Politecnico di Millano, beschikbaar op "http://iridia.ulb.ac.be/ mdorigo/ACO/publications.html", 1991.

[29] M. Dorigo, V. Maniezzo, A. Colorni. The Ant System: Optimization by a Colony of Cooperating Agents. *IEEE Transactions on Systems, Man, and Cybernetics-Part B.* Vol. 26 No. 1, pp. 29-41, 1996.

[30] D. Dubois, H. Prade. What Are Fuzzy Rules and How to Use Them. *Fuzzy Sets and Systems.* Vol. 84, pp. 169-185, 1996.

[31] C.M. Eastman, B.J. Jansen. Coverage, Relevance, and Ranking: The impact of Query Operators on Web Search Engine Results. *ACM Transactions on Information Systems.* Vol. 21, No. 4, pp. 383-411, 2003.

[32] J.B. Free. *Pheromones of social bees.* Kluwer, 1987.

[33] S. Guerin, D. Kunkle. Emergence of Constraint in Self-Organizing Systems. *Nonlinear Dynamics, Psychology, and Life Sciences.* Vol. 8 no. 2, pp. 131-146, 2004.

[34] H. Gutowitz. Complexity-Seeking Ants. *Proceedings of the 2nd European Conference on Artificial Life.* 1993.

[35] H. Gutowitz, C. Langton. Mean field theory of the edge of chaos. *Proceedings of the 3th European Conference on Artificial Life.* pp. 52-64, 1995.

[36] J. Handl. *Ant-based methods for tasks of clustering and topographic mapping: improvements, evaluation and comparison with alternative methods.* Master thesis, Friedrich-Alexander-Universität, 2003.

[37] J. Handl, B. Meyer. Improved Ant-Based Clustering and Sorting in a Document Retrieval Interface. *Proceedings of the Seventh international Conference on Parallel Problem Solving from Nature.* pp. 913-923, 2002.

[38] T. B. Ho, N. B. Nguyen, S. Kawasaki. Tolerance Rough Set Model Approach to Document Clustering. *Workshop on Text Mining, The Sixth ACM-SIGKDD International Conference on Knowledge Discovery and Data Mining.* Beschikbaar op "http://www-2.cs.cmu.edu/ dunja/WshKDD2000.html", 2000.

[39] K.M. Hoe, W.K. Lain T.S.Y. Tai. Homogeneous Ants for Web Document Similarity Modeling and Categorization. *Proceedings of Ant Algorithms : Third International Workshop.* pp. 256-261, 2002.

[40] O. Holland, C. Melhuis. Stigmergy, Self-Organization, and Sorting in Collective Robotics. *Artificial Life 5.* pp. 173-202, 1999.

[41] B. Hölldobler, E.O. Wilson. *The ants.* Springer-Verslag Heidelberg, 1990.

[42] B.J. Jansen, A. Spink. An Analysis of Web Documents Retrieved and Viewed. *Proceedings of the 4th International Conference on Internet Computing.* pp. 65-69, 2003.

[43] B.J. Jansen, A. Spink, T. Saracevic. Real life, real users, and real needs: a study and analysis of user queries on the web. *Information Processing and Management.* Vol. 36, pp. 207-227, 2000.

[44] P.M. Kanade, L.O. Hall. Fuzzy Ants as a Clustering Concept. *Proceedings of the 22nd International conference of the North American fuzzy information processing society.* pp. 227-232, 2003.

[45] R.E. Kent. Introduction to Dialectical Nets. *Proceedings of the 25th Annual Allerton Conference on Communication, Control, and Computing.* Vol. II, pp. 1204-1213, 1987.

[46] R.E. Kent. Enriched Interpretation. *Proceedings of the 3rd International Workshop on Rough Sets and Soft Computing.* pp. 116-123, 1994.

[47] R.E. Kent. Rough Concept Analysis: A Synthesis of Rough Sets and Formal Concept Analysis. *Fundamenta Informaticae.* Vol. 27, pp. 169 - 181, 1996.

[48] E.E. Kerre. *Introduction to the Basic Principles of Fuzzy Set Theory and Some of its Applications.* Communication and Cognition, Gent, 1993.

[49] E.E. Kerre. *Cursusnota's bij het opleidingsonderdeel "Vaagheids- en Onzekerheidsmodellen".* Universiteit Gent, 2003.

[50] M. Kobayashi, M. Aono. Vector Space Models for Search and Cluster Mining. *Survey of Text Mining. Clustering, Classification and Retrieval.* Springer-Verslag New York, pp. 103-121, 2003.

[51] W. Kraaij, R. Pohlmann. Porter's stemming algorithm for Dutch. *Informatiewetenschap 1994: Wetenschappelijke bijdragen aan de derde STINFON Conferentie.* pp. 167-180, 1994.

[52] K. Krishna, R. Krishnapuram. A Clustering Algorithm for Asymmetrically Related Data with Applications to Text Mining. *Proceedings of the 10th International Conference on Information and Knowledge Management.* pp. 571-573, 2001.

[53] N. Labroche, N. Monmarché, G. Venturini. AntClust: Ant Clustering and Web Usage Mining. *Proceedings of the Genetic and Evolutionary Computation Conference.* pp. 25-36, 2003.

[54] D. de Léon, J. Holsánová. Revealing User Behaviour on the World-Wide Web. *Lund University Cognitive Studies 60.* Beschikbaar op "http://www.lucs.lu.se/Abstracts/LUCS_Studies/LUCS60.html", 1996 .

[55] K. Lerman. *Document Clustering in Reduced Dimension Vector Space.* Beschikbaar op "http://www.isi.edu/ lerman/papers/Lerman99.pdf", 1999.

[56] E.D. Lumer, B. Faieta. Diversity and Adaptation in Populations of Clustering Ants. *From Animals to Animats 3: Proceedings of the 3th International Conference on the Simulation of Adaptive Behaviour.* pp. 501-508, 1994.

[57] Y.S. Maarek, R. Fagin, I.Z. Ben-Shaul, D. Pelleg. Ephemeral Document Clustering for Web Applications. IBM Research Report RJ 10186. Beschikbaar op "http://www.almaden.ibm.com/cs/people/fagin/cluster.pdf", 2000.

[58] M. Martin, B. Chopard, P. Albuquerque. Formation of an Ant Cemetery: Swarm Intelligence or statistical accident? *Future Generation Computer Systems.* Vol. 18 no. 7, pp. 951-959, 2002.

[59] N. Monmarché. *Algorithmes de fourmis artificielles: applications à la classification et à l'optimisation.* Doctoraatsproefschrift, Université François Rabelais, 2000.

[60] N. Monmarché, M. Slimane, G. Venturini. *AntClass: discovery of clusters in numeric data by a hybridization of an ant colony with the Kmeans algorithm.* Internal report no. 213, 1999.

[61] S. Osiński. *An Algorithm for Clustering of Web Search Results.* Master thesis, Poznań University of Technology, 2003.

[62] L. Page, S. Brin, R. Motwani, T. Winograd. *The PageRank Citation Ranking: Bringing Order to the Web.* Stanford Digital Libraries SIDL-WP-1999-0120, beschikbaar op "http://dbpuls.stanford.edu/pub/1999-66", 1998.

[63] C.H. Papadimitriou, P. Raghavan, H. Tamaki, S. Vempala. Latent Semantic Indexing: A Probabilistic Analysis. *Proceedings of the ACM Conference on Principles of Database Systems.* pp. 159-168, Seattle, 1998.

[64] R. S. Parpinelli, H. S. Lopes, A. A. Freitas. Data Mining with an Ant Colony Optimization Algorithm. *IEEE Transactions on Evolutionary Computation.* Vol. 6 no. 4, pp. 321-332, 2002.

[65] Z. Pawlak. Rough Sets. *International Journal of Information and Computer Science.* Vol. 11, pp. 341-356, 1982.

[66] R. Puystjens. *Cursusnota's bij het opleidingsonderdeel Lineaire Algebra.* Universiteit Gent, 2000.

[67] V. Ramos, J.J. Merelo. Self-Organized Stigmergic Document Maps: Environment as a Mechanism for Context Learning. *Proceedings of the 1st Spanish Conference on Evolutionary and Bio-Inspired Algorithms.* pp. 284-293, 2002.

[68] V. Ramos, F. Muge, P. Pina. Self-Organized Data and Image Retrieval as a Consequence of Inter-Dynamic Synergistic Relationships in Artificial Ant Colonies. *Soft Computing Systems: Design, Management and Applications.* pp. 500-509, 2002.

[69] E.M. Rauch, M.M. Millonas, D.R. Chialvo. Pattern Formation and Functionality in Swarm Models. *Physics Letters A.* Vol. 207 no. 3-4, pp. 185-193, 1995.

[70] A.M. Radzikowska, E.E. Kerre. A comparative study of fuzzy rough sets. *Fuzzy Sets and Systems.* Vol. 126, pp. 137-156, 2002.

[71] A. Strehl, J. Ghosh, R. Mooney. Impact of Similarity Measures on Web-page Clustering. *Proceedings of the AAAI Workshop on AI for Web Search.* pp. 58-64, 2000.

[72] D.J.T. Sumpter, M. Beekman. From non-linearity to optimality: pheromone trail foraging by ants. *Animal Behaviour.* Vol. 66, pp. 273-280, 2003.

[73] D.J.T. Sumpter, S.C. Pratt. A framework for modelling social insect foraging. *Behavioural Ecology and Sociobiology.* Vol. 53, pp. 131-144, 2003.

[74] G. Theraulaz, E. Bonabeau, J.L. Deneubourg. Response threshold reinforcement and division of labour in insect societies. Working Paper 98-01-006, Santa Fe Institute, beschikbaar op: http://ideas.repec.org/p/wop/safiwp/98-01-006.html, 1998.

[75] G. Theraulaz, E. Bonabeau, S. C. Nicolis, R.V. Solé, V. Fourcassié, S. Blanco, R. Fournier, J.L. Joly, P. Fernández, A. Grimal, P. Dalle. Spatial patterns in ant colonies. *Proceedings of the National Academy of Sciences of the Unites States of America.* Vol. 99 no. 15, pp. 9645-9649, 2002.

[76] H. Thiele. Fuzzy Rough Sets versus Rough Fuzzy Sets - An Interpretation and a Comparative Study using Concepts of Modal Logics. *5th European Congress on Intelligent Techniques & Soft Computing.* Vol. 1, pp. 159-167, 1997.

[77] E. Tsiporkova, H.-J. Zimmermann. Aggregation of Compatibility and Equality: A New Class of Similarity Measures for Fuzzy Sets. *Proceedings of the 7th International Conference on Information Processing and Management of Uncertainty in Knowledge-based Systems.* pp. 1769-1776, Paris, 1998.

[78] Cheng-Fa Tsai, Han-Chang Wu, Chun-Wei Tsai. A New Data Clustering Approach for Data Mining in Large Databases. *Proceedings of the International Symposium on Parallel Architectures, Algorithms and Networks.* pp. 315-320, 2002.

[79] S. Van Aelst. *Cursusnota's bij het opleidingsonderdeel "Statistische analyse van data".* Universiteit Gent, 2003.

[80] D. Weiss. *Carrot^2: Developers guide.* 2002.

[81] J. Stefanowski, D. Weiss. Carrot^2 and Language Properties in Web Search Results Clustering. *Advances in Web Intelligence, Proceedings of the First International Atlantic Web Intelligence Conference.* Vol. 2663, pp. 240-249, 2002.

[82] R. Wille. Restructuring Lattice Theory: An Approach Based on Hierarchies of Concepts. *Ordered Sets.* pp. 445-470, Reidel, Dordrecht-Boston, 1982.

[83] Y.Y. Yao, T.Y. Lin. Generalisation of Rough Sets using Modal Logics. *Intelligent Automation and Soft Computing, An International Journal.* Vol. 2 no. 2, pp. 103-120, 1996.

[84] O. Zamir, O. Etzioni. Web Document Clustering: A Feasibility Demonstration. *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval.* pp. 46-54, 1998.

[85] O. Zamir, O. Etzioni. Grouper: A Dynamic Clustering Interface to Web Search Results *Computer Networks*. Vol. 31 no. 11-16, pp. 1361-1374, 1999.

[86] D. Zhang, Y. Dong. Semantic, Hierarchical, Online Clustering of Web Search Results. *Proceedings of the 6th Asia Pacific Web Conference*. pp. 69-78, 2004.