

# 9

## Evaluatie

In dit hoofdstuk zullen wij een korte evaluatie geven van het statistisch patroonherkennen. Op grond van een verzameling *leerobjecten* met *bekende classificatie*

$$\mathcal{L} = \{(\mathbf{x}_i, \lambda_i); i = 1, \dots, N\}$$

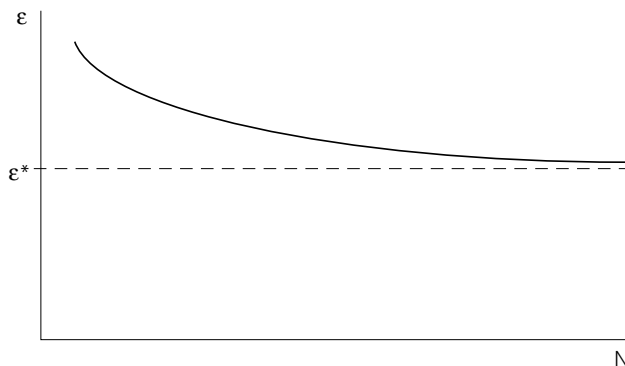
met  $\lambda_i = A, B$  als klasselabels en  $N$  als het aantal leerobjecten, zijn schattingen  $\hat{f}_A(\mathbf{x})$  en  $\hat{f}_B(\mathbf{x})$  gemaakt van de kansverdelingen  $f_A(\mathbf{x})$  en  $f_B(\mathbf{x})$ . Hierbij is gebruik gemaakt van bekende of vermeende eigenschappen van deze dichtheden en de kenmerken  $x_1, x_2, \dots, x_k$ . De uiteindelijk gevonden scheidingsfunctie kan onder andere worden geschreven als

$$S(\mathbf{x}) = \log(P_A \hat{f}_A(\mathbf{x})) - \log(P_B \hat{f}_B(\mathbf{x})).$$

Er zijn schattingsmethoden aangegeven voor de kans op foutieve classificaties:

$$\varepsilon = \text{Prob}\{S(\mathbf{x}) < 0 \mid \mathbf{x} \in A\}P_A + \text{Prob}\{S(\mathbf{x}) \geq 0 \mid \mathbf{x} \in B\}P_B.$$

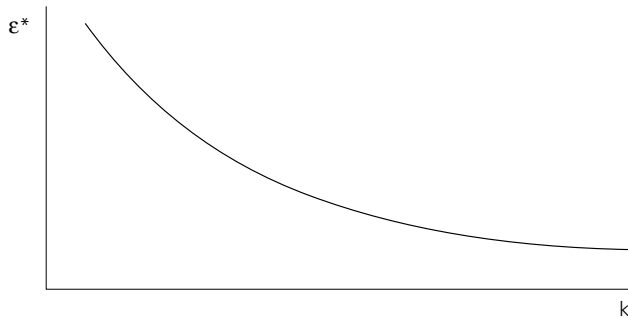
Wij hebben gezien dat voor geschikt gekozen methoden  $\varepsilon$  als functie van  $N$  daalt naar de Bayes-fout  $\varepsilon^*$ , zie figuur 9.1. Dit is de fout die wordt bereikt bij bekende  $f_A(\mathbf{x})$  en  $f_B(\mathbf{x})$ . Wij kunnen stellen dat voor  $N \rightarrow \infty$  in feite, hoewel wellicht niet expliciet en analytisch, deze dichtheden bekend zijn.



**Figuur 9.1.** De foutkans daalt als functie van het aantal leerobjecten naar de Bayes-foutkans.

Er zijn zowel zuivere als onzuivere methoden aangegeven om de foutkans  $\varepsilon$  te schatten op grond van de leerverzameling. In het algemeen hadden deze methoden de eigenschap dat voor  $N \rightarrow \infty$  zowel de onzuiverheid als de variantie van de schatter naar nul naderen.

In hoofdstuk 7 is reeds aandacht gegeven aan het gedrag van de foutkans als functie van het *aantal* kenmerken. Wij zullen hierop in dit hoofdstuk nogmaals de aandacht vestigen. In het algemeen kunnen wij stellen dat door het toevoegen van kenmerken de kansdichtheidsverdelingen alleen maar verder uit elkaar kunnen komen te liggen.  $\varepsilon^*(k)$  zal dus een *monotoon dalende functie* zijn. Immers, heeft men een kenmerk gevonden dat geschikt is om de klassen te onderscheiden, dan kan dit door het in beschouwing nemen van nog zoveel andere kenmerken niet ongedaan worden gemaakt. Zie figuur 9.2.

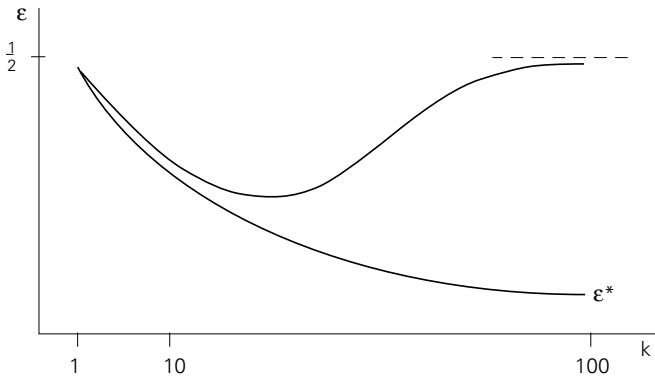


**Figuur 9.2.** De Bayes-foutkans is een monotoon dalende functie van het aantal kenmerken.

De Bayes-fout  $\varepsilon^*(k)$  hoeft niet noodzakelijk naar nul te naderen voor  $k \rightarrow \infty$ . Het is mogelijk dat zelfs bij het in beschouwing nemen van alle mogelijke kenmerken er nog steeds objecten zijn die fout worden ingedeeld, bijvoorbeeld doordat sommige klassen volledig identieke objecten kunnen bevatten. Zo kan het karakter “0” zowel een nul als de letter O zijn.

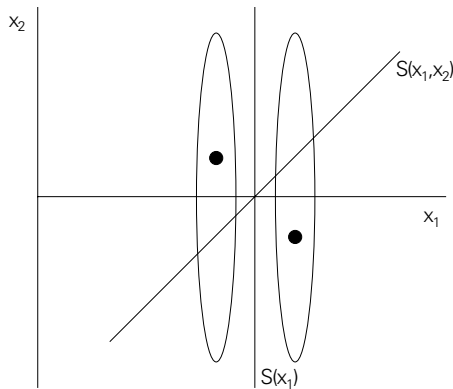
Het bovenstaande houdt nog geen rekening met de gevolgen van een *eindige grootte* van de leerverzameling. In samenhang met de dimensionaliteit, het aantal kenmerken, kan het volgende gebeuren. Een leerverzameling van 50 objecten kan een één-dimensionale scheidingsfunctie ( $k=1$ ) redelijk nauwkeurig schatten:  $\varepsilon(1) \approx \varepsilon^*(1)$ . Bij verhoging van  $k$  naar 10 zal echter een onnauwkeurigheid optreden in de schatting van de dichtheden:  $\varepsilon(10) > \varepsilon^*(10)$ . Voor zeer hoge dimensionaliteit, bijvoorbeeld  $k = 100$ , is het in het geheel niet mogelijk om tot goede schattingen te komen. De gevonden scheiding wordt betekenisloos:  $\varepsilon(100) = 0.5$  (zie figuur 9.3).

Het verschijnsel hier staat bekend als het *piekeffect*. Het wordt veroorzaakt doordat uiteindelijk bij toenemend aantal kenmerken (en dus steeds hogere dimensionaliteit) de steeds groter wordende schattingsfouten het winnen van de extra scheidingsmogelijk-



**Figuur 9.3.** *Het piekeffect: de foutkans als functie van het aantal kenmerken.*

heden van het nieuwe kenmerk. Dit is niet onvoorwaardelijk waar. Het is afhankelijk van de reeds bereikte scheiding en de ruis in nieuwe kenmerken. Een zeer eenvoudig voorbeeld is gegeven in figuur 9.4.



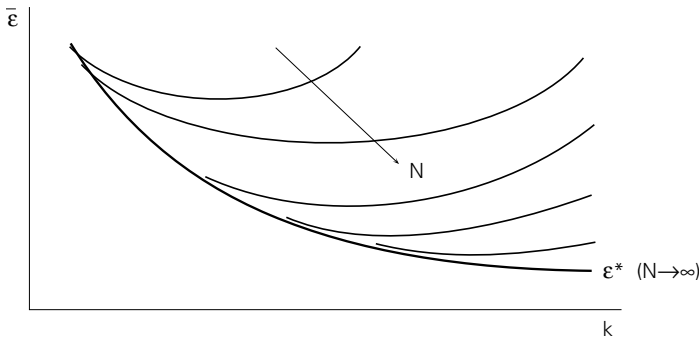
**Figuur 9.4.** *Gebruik van twee kenmerken leidt bij twee leerobjecten tot een slechter resultaat dan met één kenmerk.*

Met ieder tweetal leerobjecten, één uit  $A$  en één uit  $B$ , wordt een zeer goede scheiding bereikt uitsluitend op grond van  $x_1$  (hier is de middelloodlijn genomen). Wordt echter ook het zinloze kenmerk  $x_2$  gebruikt dan zal vrijwel ieder tweetal leerobjecten tot een veel slechtere scheidingsfunctie leiden.

Niet bij ieder tweetal leerobjecten zal verslechtering optreden. In dit voorbeeld is dit echter wel de verwachting, d.w.z. dat

$$\bar{\varepsilon} = E_L\{\varepsilon\}$$

als de verwachte foutkans in relatie tot de grootte van de leerverzameling het piekeffect vertoont. De optimale waarde van  $k$  waarna verslechtering optreedt is afhankelijk van de grootte van  $N$  (zie figuur 9.5).



**Figuur 9.5.** De verwachte classificatiefout als functie van het aantal leerobjecten en het aantal kenmerken.

Bij toenemende  $N$  is het dus verantwoord om steeds meer kenmerken te gebruiken. Exacte waarden van  $k$  en  $N$  zijn echter in algemeenheid niet te geven. *Vuistregels* zijn dat er niets meer te verwachten is voor  $k > N/m$ , waarbij  $m$  het aantal klassen is. Voor een twee-klassenprobleem wordt er in de regel naar gestreefd om  $k > N/10$  te houden. Van geval tot geval kunnen de optimale waarden echter anders liggen. Het ernstige probleem dat zich hierbij voordoet is dat in de praktijk, op grond van een leerverzameling, slechts een schatting  $\hat{\epsilon}$  van  $\epsilon$  is te verkrijgen. Het detecteren van pieking van  $\epsilon$  op grond van  $\hat{\epsilon}(k)$  is statistisch vrijwel uitgesloten. Er zullen immers alleen kenmerken worden geselecteerd waarvoor  $\hat{\epsilon}(k)$  daalt. Uiteindelijk zal dit uitsluitend op grond van statistische fluctuaties gebeuren, waardoor nieuwe kenmerken niet wezenlijk beter hoeven te zijn. Detectie hiervan zou mogelijk zijn wanneer er behalve een nieuw kenmerk ook een nieuwe verzameling objecten zou zijn om het kenmerk in samenhang met de reeds geselecteerde kenmerken te evalueren. In de praktijk is dit niet mogelijk, zodat steeds dezelfde leerverzameling moet worden gebruikt met het risico van pieking.

Op grond van het piekeffect moet worden geconcludeerd dat kenmerken liefst niet (uitsluitend) op grond van de statistiek moeten worden geselecteerd maar (zo mogelijk) op grond van andere kennis over het herkenningprobleem. Vanuit het statistisch patroonherkennen gezien is dit dan *a priori kennis*. De vraag luidt dan hoe deze kennis dan ooit verworven kan zijn als de statistiek als bron van kennis is uitgesloten. Dit leidt dan tot de *epistemologie*, de *kennisleer*, een tak van de filosofie. Deze stelt dat kennis tot stand komt op grond van *autoriteit* (overlevering), *denken* of *waarneming*. Met een uitwerking van deze laatste kennisbron hebben wij ons in het voorgaande bezig

gehouden. De eerste kennisbron, de autoriteit, is op zich voor de mensheid als geheel geen bron van kennis. In engere zin echter vormt de autoriteit (*expert*) in relatie tot een herkenningsprobleem echter wel een kennisbron. In deze zin zullen wij *kennisgestuurde* oplossingen voor herkenningsproblemen beschouwen. Tenslotte het denken (*redeneren?*). Hoewel een uitwerking van denken buiten de materie van dit boek valt, is kunstmatig redeneren – als probleem uit de *kunstmatige intelligentie* – veelal besloten in *kennisverwerkende systemen (expertsystemen)*.

Aan het eind van deze zuivere statistische benadering van het herkenningsprobleem keren wij nog eenmaal terug naar de uitgangspunten. Er zijn daar enige mogelijkheden voor alternatieven gepasseerd die tot nu toe onbesproken zijn gebleven. Deze zullen hier kort worden genoemd en in de volgende hoofdstukken worden uitgewerkt. Achtereenvolgens zullen ter sprake komen de onderwerpen: *klassevorming*, de *klasselabels*, *subjectieve kenmerken* en het *statistisch model*.

Objecten met hetzelfde klasselabel zijn behandeld als één enkele klasse. Het is echter zeer wel denkbaar dat zo'n klasse beter zou kunnen worden beschreven met behulp van een aantal subklassen. Zo bestaat bijvoorbeeld de klasse van abnormale electrocardiogrammen uit een aantal subklassen die corresponderen met specifieke hartafwijkingen. Deze kunnen ieder op zich redelijk van de klasse van de normale electrocardiogrammen worden gescheiden omdat zo'n subklasse vrij homogeen kan zijn. Het oorspronkelijk scheidingsprobleem is echter complex vanwege de inhomogeniteit van de klasse van abnormalen als geheel. De analyse van zo'n klasse naar het bestaan van homogene subklassen heet clusteranalyse. Hierin wordt dus geprobeerd tot onderscheid te komen van groepen objecten met oorspronkelijk hetzelfde label.

De klasselabels zoals ze zijn behandeld zijn “*harde*” labels. Objecten behoorden òf tot klasse *A* òf tot klasse *B*. Een tussenweg was er niet. In de praktijk is het echter zeer wel denkbaar dat diegene die de labels heeft toegekend wel degelijk *twijfel* heeft ervaren tussen de mogelijkheden *A* of *B*. Ook zou hij *kennis* kunnen hebben over het feit dat een bepaald object duidelijk tot klasse *A* behoort, maar binnen deze klasse eigenlijk een uitzonderlijk exemplaar is. Om deze vormen van kennis te kunnen benutten worden de harde labels wel vervangen door “*zachte*” labels (*vage* of *fuzzy* labels). Een object krijgt voor iedere klasse een fuzzy label, een getal tussen nul en een, welke de *mate van lidmaatschap* tot de betreffende klasse tot uitdrukking brengt. Deze beschrijvingswijze laat in principe toe dat een object voor meer klassen een lidmaatschapswaarde in de buurt van een krijgt, dan wel voor alle klassen een lidmaatschapswaarde van bijna nul.

Naast de labels kunnen ook de kenmerken gebaseerd zijn op een *subjectieve beoordeling*. Hiervoor zouden ook fuzzy labels kunnen worden benut.

Met betrekking tot het in de vorige hoofdstukken gehanteerde *model* kunnen twee opmerkingen gemaakt worden:

- 1.** Het is *statistisch* van aard; dit houdt in dat kennis, welke geen betrekking heeft op, of om te vormen is tot kansen en/of kansdichtheden, niet gebruikt kan worden. *Structurele* of *formele* kennis over samenhang tussen objecten en kenmerken kan op andere wijzen worden beschreven. Hierbij is echter weer vaak het probleem hoe een verzameling leerobjecten statistisch kan worden benut.
- 2.** Van de verzameling leerobjecten wordt verondersteld dat het een *aselecte trekking* is uit het universum van objecten. In sommige gevallen is dit echter onjuist en zijn de leerobjecten op een andere wijze geselecteerd. Het gehele probabilistische raamwerk is dan onbruikbaar. Er moet dan worden teruggevallen op de *verzamelingenleer* (*possibilistisch* in plaats van *probabilistisch*). In samenhang met de eerder genoemde fuzzy labels kunnen dan echter nog bruikbare systemen worden ontwikkeld. Een en ander zal in de volgende hoofdstukken worden uitgewerkt.

# 10

## Syntactisch patroonherkennen

We hebben gezien dat bij het statistisch patroonherkennen objecten worden gekarakteriseerd door een verzameling kenmerken: *meetwaarden* van karakteristieke eigenschappen. Relaties tussen kenmerken konden slechts op een statistische wijze worden beschreven. Bij het *syntactisch patroonherkennen* worden objecten beschreven met *primitieven*. De *structuur* van objecten, dat wil zeggen de relaties tussen primitieven, wordt formeel beschreven. De structuur van objecten van dezelfde klasse wordt vaak gedefinieerd door een *grammatica* waarin de primitieven de *symbolen* zijn. Verschillende klassen hebben verschillende grammatica's. Met behulp van leerobjecten kunnen grammatica's worden geleerd ("*geïnfereerd*"). Objecten van een onbekende klasse kunnen worden geclassificeerd door te bepalen aan welke grammatica(s) ze voldoen ("*parsing*"). Hieronder zullen we op grammatica's, parsen en infereren nader ingaan. Allereerst zullen echter de verschillen met het statistisch patroonherkennen worden besproken.

Primitieven zijn grootheden die wel of niet aanwezig zijn. Wat van object tot object vooral verschilt is de wijze waarop de primitieven samen het object structureren. Binnen het syntactisch patroonherkennen worden objecten dus in eerste instantie *logisch* beschreven. Binnen het statistisch patroonherkennen gebeurt dat met vectoren van kenmerkwaarden. De klassebeschrijving is in het ene geval door middel van grammatica's, in het andere door middel van kansdichtheden. Syntactisch beschreven objecten voldoen dus wel of niet aan een grammatica en kunnen eventueel daardoor tot meer dan één of tot geen enkele klasse behoren. Statistisch beschreven objecten behoren daarentegen met een zekere waarschijnlijkheid tot een bepaalde klasse. Het gevolg hiervan is dat ruis en onzekerheid moeilijk syntactisch doch goed statistisch kunnen worden beschreven.

---

Notaties hoofdstuk 10

$S$	startsymbool	$\mathcal{P}$	verzameling produktieregels
$\mathcal{V}_N$	verzameling hulpsymbolen	$\mathcal{G}$	grammatica
$\mathcal{V}_T$	verzameling eindsymbolen		

Samenvattend biedt het syntactisch patroonherkennen vooral een logische, structurele beschrijving en het statistisch patroonherkennen een onderscheid op basis van waarschijnlijkheden.

Met een grammatica worden objecten met behulp van *herschrijfregels* in steeds meer elementaire stukken opgedeeld. Bijvoorbeeld een stad kan worden beschreven met de volgende reeks van regels:

- |                             |   |                           |
|-----------------------------|---|---------------------------|
| 1. stad                     | → | stad    wijk              |
| 2. stad                     | → | centrum    wijk           |
| 3. wijk                     | → | wijk    park              |
| 4. wijk                     | → | wijk    straat            |
| 5. wijk                     | → | wijk    laan              |
| 6. wijk    straat    straat | → | plein    straat    straat |
| 7. wijk    park             | → | wijk    laan    park      |
| 8. straat                   | → | weg    huizen             |
| 9. laan                     | → | weg    bomen              |
| 10. laan                    | → | straat    bomen           |
| 11. park                    | → | park    bos               |
| 12. park                    | → | gras    paden    perken   |
| 13. bos                     | → | bos    boom               |
| 14. bos                     | → | boom    boom    boom      |

De symbolen die in deze herschrijfregels (*produktieregels*) worden gebruikt kunnen in drie groepen worden ingedeeld:

1. Het *startsymbool*  $S$ , hier  $S = \text{stad}$ .
2. De verzameling van *hulpsymbolen*  $\mathcal{V}_N$  (non-terminals).  
Hier is  $\mathcal{V}_N = \{\text{wijk, straat, laan, park, bos}\}$ .
3. De verzameling van *eindsymbolen*  $\mathcal{V}_T$  (terminals).  
Hier is  $\mathcal{V}_T = \{\text{centrum, plein, weg, huizen, boom, gras, paden, perken}\}$ .

Tesamen met de verzameling produktieregels  $\mathcal{P}$  wordt de grammatica nu volledig gedefinieerd door:

$$\mathcal{G} = \{S, \mathcal{V}_N, \mathcal{V}_T, \mathcal{P}\}.$$

De produktieregels kunnen in een willekeurige volgorde worden afgelopen, mits maar met het startsymbool wordt begonnen en de start- en hulpsymbolen uiteindelijk alle worden herschreven tot eindsymbolen. Er zijn daardoor vele steden mogelijk. De grammatica legt echter ook duidelijk beperkingen op. Zo bestaat de kleinste stad uit één centrum en minstens één wijk en zijn de regels zo dat een park óf geen óf drie óf meer bomen heeft.

Door Chomsky is een *hiërarchische* indeling van talen gemaakt aan de hand van verschillende *typen* van produktieregels.

*Type 0:* Alle produktieregels zijn van het type:

$$\alpha \rightarrow \beta, \quad \text{met } \alpha, \beta \in \mathcal{V}_N \cup \mathcal{V}_T.$$

$\alpha$  en  $\beta$  kunnen ook rijen van eind- en hulpsymbolen zijn. Er zijn geen beperkingen. Dit type omvat dus alle grammatica's die met produktieregels zijn te schrijven. Er zijn echter talen (*verzamelingsen symboolrijen*) die hier nog buiten liggen en die dus grammatica's hebben die niet met een dergelijk stelsel van produktieregels zijn te schrijven.

*Type 1:* Als alle produktieregels van het type:

$$\alpha_1 A \alpha_2 \rightarrow \alpha_1 \beta \alpha_2, \quad \text{met } A \in \mathcal{V}_N \quad \text{en} \quad \alpha_1, \alpha_2, \beta \in \mathcal{V}_N \cup \mathcal{V}_T$$

zijn, dan heet de bijbehorende taal (grammatica) *context gevoelig*.

*Type 2:* Een deelverzameling van de contextgevoelige talen zijn de contextvrije talen met alleen regels als:

$$A \rightarrow \beta, \quad \text{met } A \in \mathcal{V}_N, \beta \in \mathcal{V}_N \cup \mathcal{V}_T.$$

Dus:

*Type 3:* Een deelverzameling van de contextvrije talen zijn de *reguliere talen*.

Hiervan zijn er twee typen:

$$\text{a. } \left\{ \begin{array}{l} \text{of } A \rightarrow aB \\ A \rightarrow a \end{array} \right., \quad A, B \in \mathcal{V}_N, \quad a \in \mathcal{V}_T.$$

$$\text{b. } \left\{ \begin{array}{l} \text{of } A \rightarrow Ba \\ A \rightarrow a \end{array} \right., \quad A, B \in \mathcal{V}_N, \quad a \in \mathcal{V}_T.$$

Het aantal *zinnen* (symboolrijen) dat met de meest beperkte grammatica kan worden gemaakt, de reguliere, is al *oneindig* groot.

Zo genereert

$$S \rightarrow aS$$

$$S \rightarrow a$$

zinnen  $a^n$  met  $n \geq 1$ .

Iedere reguliere grammatica komt overeen met een *graaf* met een *eindig aantal toestanden*. Iedere produktieregel definieert een overgang tussen twee toestanden. Een *toestandsmachine* die volgens deze graaf werkt is in principe in staat om van een zin (reeks symbolen) vast te stellen of deze tot de bijbehorende grammatica behoort. Dit

gebeurt door volgens de opeenvolgende symbolen de toestanden af te lopen. Als na het laatste symbool een zogenaamde eindtoestand is bereikt dan is de zin geaccepteerd en voldoet deze dus aan de betreffende grammatica. Anders moet de zin worden verworpen.

Bij de andere grammatica's bestaan meer gecompliceerde machines om de bijbehorende zinnen te parsen. Zo kan een contextvrije taal worden geparsed door een "push-down"-automaat, een contextgevoelige taal door een *lineair begrensde automaat* en een type 0 taal door een *Turing machine*.

In al deze gevallen wordt er uitgegaan van *exacte herkenning*, dat wil zeggen als de zin exact volgens de betreffende grammatica kan worden gegenereerd dan wordt hij geaccepteerd, anders wordt hij verworpen. Een mogelijkheid om ruis toe te laten is het definiëren van *afstandsmaten* tussen mogelijke en aangeboden zinnen. De "afstand" tussen een zin en een grammatica kan dan worden gedefinieerd als de kleinste afstand tussen die zin en alle zinnen die door die grammatica kunnen worden gegenereerd. Moet een zin worden ingedeeld in een van de mogelijke grammatica's dan kan hij worden toegewezen tot die grammatica waar hij de kleinste afstand toe heeft. De afstand tussen twee zinnen kan bijv. worden gedefinieerd als de som van de kosten verbonden aan *substituties*, *weglatingen* en *aanvullingen* van enkele symbolen, zodanig dat deze som *minimaal* is en de ene zin overgaat in de andere zin.

Tenslotte nog iets over het vinden van grammatica's op grond van een verzameling zinnen uit een taal (leerobjecten). Dit heet *infereren*. Het inferentieproces start bijvoorbeeld met een aantal produktieregels waarin steeds het startsymbool als een zin corresponderend met een leerobject wordt herschreven. In volgende stappen worden regels gecombineerd, eventueel nadat ze gesplitst zijn in gemeenschappelijke stukken. Op dit moment genereert de grammatica nog exact de leerverzameling en geen zin meer. In een volgende fase worden er veralgemeniseringen ingevoerd waarbij door samenvoegingen en eventueel *recursieve* regels ineens veel meer zinnen kunnen worden geproduceerd. Dit is een kritiek moment. De vraag is welke veralgemeniseringen toelaatbaar zijn en welke niet. Dit moet bepaald worden door de voorkennis die er met betrekking tot het probleem bestaat. Net als bij het statistisch patroonherkennen vinden we hier dat binnen het formele leerschema geen volledige oplossing wordt gevonden en dat extra kennis van buitenaf nodig is om tot een bevredigend resultaat te komen.

# 11

## Clusteranalyse

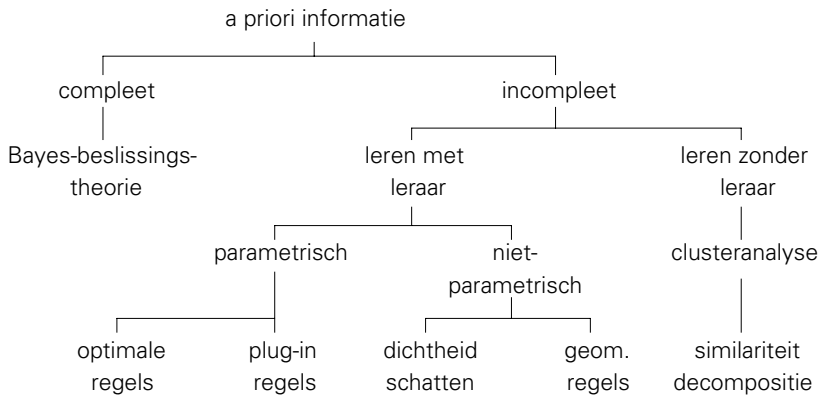
Patroonherkenning hebben wij leren kennen als een verzameling van rekenkundige technieken voor de analyse en interpretatie van multivariate data; in brede zin behoort de *clusteranalyse* ook tot deze technieken. In beginsel is clusteranalyse een discipline waarin objecten gegroepeerd worden op basis van *indices* die de *mate van overeenkomst* uitdrukken. Het doel van de patroonherkenning is het op een reken-technisch aantrekkelijke wijze beslissen over een categorie(klasse-)label dat aan het object wordt toegekend. Gegeven een set van leerobjecten (objecten waarvan het klaslabel a priori bekend is), verschaft de patroonherkenning een scala aan algoritmen om de patroonruimte op te delen in deelruimten overeenkomend met de objectklassen. De mate waarin a priori kennis aanwezig is bepaalt in hoge mate het ontwerp van de classificator. Zo hebben wij gezien dat, indien de klasconditionele kansdichtheden a priori gegeven zijn, optimale beslissingsregels kunnen worden verkregen met behulp van de statistische beslissingstheorie en discriminant analyse. Indien slechts leerobjecten op zich gegeven zijn, dan is de patroonruimte op te delen door middel van lineaire of kwadratische scheidingsfuncties of door gebruik te maken van naaste-nabuur beslissingsregels. De hierboven gegeven samenhang is geïllustreerd in figuur 11.1.

Filosofisch bezien schuilt het grote verschil tussen patroonherkenning en clusteranalyse in de rol die klasselabels hierbij spelen. De klasselabels waren cruciaal voor het ontwerpen van beslissingsregels. In de clusteranalyse vormen ze hoogstens een verificatiemiddel voor de uiteindelijk gevonden groepen. Met andere woorden: patroonherkenning gebruikt extrinsieke informatie en clusteranalyse gebruikt slechts intrinsieke informatie. Het onderscheid wordt ook wel geformuleerd als “*leren met leraar*” (patroonherkenning) en “*leren zonder leraar*” (clusteranalyse; *unsupervised learning*).

---

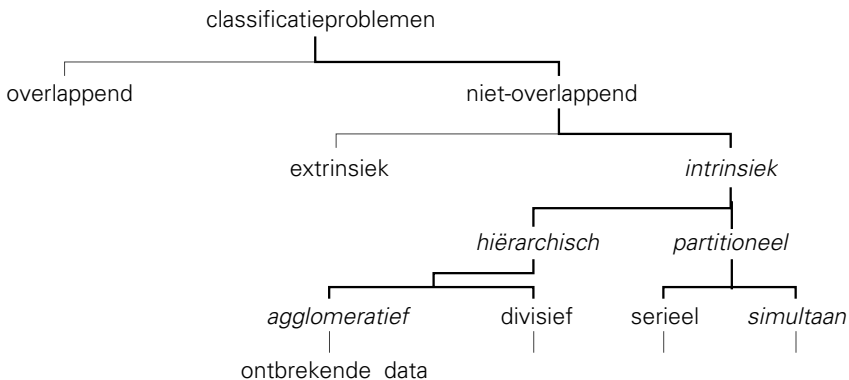
### Notaties hoofdstuk 11

$T$	clustertransformatie	$N$	aantal objecten
$D$	afstands(dissimilarity)matrix	$m$	aantal clusters
$d(i,j)$	afstand tussen $\mathbf{x}_i$ en $\mathbf{x}_j$	$\mathcal{E}_k^2$	kwadratisch foutcriterium voor $k$ -de cluster
CPCC	CoPhenetic Correlation Coefficient	$E_m^2$	kwadratisch foutcriterium voor $m$ -partitie
$L_i$	$i$ -de clusterniveau		
$C_i$	$i$ -de cluster		



**Figuur 11.1.** Probleemindeling in de patroonherkenning.

De relatie tussen beide disciplines is gelegen in het feit dat het bepalen van “natuurlijke” objectgroeperingen vooraf kan gaan aan het formuleren van beslissingsfuncties. Beide disciplines zijn op gelijke wijze behept met het dimensionaliteitsprobleem, terwijl het aantal in de beschouwing mee te nemen objecten theoretisch niet groot genoeg, maar in de praktijk om rekentechnische redenen beperkt zal moeten blijven.



**Figuur 11.2.** Classificatie van clusterproblemen.

De clustermethoden laten zich classificeren volgens figuur 11.2. Wij onderscheiden bij de algoritmen twee hoofdtypen: *hiërarchisch* en *partioneel*. Een hiërarchische classificatie is een *geneste reeks van groeperingen* terwijl een partionele classificatie een *particuliere objectindeling* beschrijft. Met andere woorden: een hiërarchische classificatie is een geneste reeks van partionele classificaties. Bij de hiërarchische methoden is een verder onderscheid: de *agglomeratieve* procedures (er wordt begonnen met kleine groepjes welke door bijeenvoegen aanleiding geven tot grote groepen) en de *divisieve* methoden (er wordt begonnen met een grote groep welke door deling steeds uiteenvalt in kleinere groepjes). Voor de hiërarchische methoden moet de data in de

vorm van *object-object-relaties* worden aangeboden (paarsgewijs), terwijl voor de partitionele methode de data in de vorm van een *objectmatrix* (ieder object met expliciet al zijn kenmerkwwaarden) gegeven moet zijn.

In wat volgt bespreken wij de hiërarchische clustermethoden en vervolgens de partitionele clustermethoden, daarbij uitsluitend ingaande op de hoofdfilosofie van de algoritmen. Voor een uitgebreidere bespreking wordt naar de literatuurlijst verwezen.

### 11.1. Hiërarchische methoden

Zoals gezegd is een hiërarchische methode gericht op een procedure voor het creëren van een *geneste reeks partities* op basis van een *relationele* inputmatrix (“*proximity matrix*”). De procedure transformeert de proximity matrix naar een zogenaamd *dendogram* (een boom welke de hiërarchische structuur representeert). De procedure start met ieder object in een aparte cluster en vervolgt door paarsgewijs clusters samen te voegen op basis van objectovereenkomst (“*similarity, resemblance*”). Dit is een *agglomeratieve* procedure. De mathematische formulering is als volgt.

Beschouw de clusterprocedure als een transformatie  $T$  met de eigenschap dat de reflexieve, symmetrische input-“dissimilarity”-matrix wordt getransformeerd in een reflexieve, symmetrische en transitieve output-matrix, zodanig dat het “verschil” tussen input- en outputmatrix minimaal is.

Uitgaande van een dissimilaritymatrix (*afstandsmatrix*) zijn de condities:

- reflexiviteit:  $d(i,i) = 0$  voor alle  $\mathbf{x}_i$ ;
- symmetrie:  $d(i,j) = d(j,i)$  voor alle  $\mathbf{x}_i$  en  $\mathbf{x}_j$ ;
- transitiviteit:  $d(i,j) < \max[d(i,k), d(k,j)]$  voor alle  $\mathbf{x}_i, \mathbf{x}_j, \mathbf{x}_k$ .

Het inbrengen van de transitiviteit vereist dat verschillende waarden in de inputmatrix mogelijk zullen moeten worden aangepast. Deze aanpassing wordt bewerkstelligt door de clusterprocedure. Het “verschil” tussen input en output is te definiëren als *de mate van correlatie* tussen beide matrices.

Hiervoor gebruikt men de zogenaamde *CoPhenetic Correlation Coefficient*, de CPCC:

$$\text{CPCC} = \frac{\frac{1}{n} \sum_{i < j} d(i,j)d'(i,j) - (\bar{d})(\bar{d}')}{\sqrt{\frac{1}{n} \sum_{i < j} d(i,j)^2 - \bar{d}^2} \sqrt{\frac{1}{n} \sum_{i < j} d'(i,j)^2 - \bar{d}'^2}},$$

met  $\bar{d} = (1/n) \sum_{i < j} d(i,j)$ ,  $\bar{d}' = (1/n) \sum_{i < j} d'(i,j)$  en  $n = N(N-1)/2$ .

Het klassieke algoritme (van Johnson), uitgaande van een dissimilaritymatrix, luidt dan:

*Stap 1*

Initialiseer door ieder object aan een unieke cluster toe te wijzen; noem het niveau hiervan  $L_0$ ; de *initiële* clustering is  $C_0 = \{(1),(2),\dots,(N)\}$ ; de index van de clustering is  $k$  (hier  $k = 0$ ).

*Stap 2*

$k = k + 1$ ;

- zoek de kleinste ingang in de matrix  $D$ ;  $d(i,j) = \min [d(s,t)]$ ;
- voeg clusters  $i$  en  $j$  bij elkaar in een nieuwe clustering  $C_k$  met  $(N - k)$  clusters;
- $L_k = d(i,j)$ .

*Stap 3*

Herschrijf  $D$  op de volgende wijze:

$$d[(k),(j)] = \min (d[(k),(i)],d[(k),(j)]), \quad k \neq i,j;$$

$$d[(k),(j)] = d[(j),(k)] \quad (\textit{Single link});$$

of

$$d[(k),(j)] = \max (d[(k),(i)],d[(k),(j)]), \quad k \neq i,j;$$

$$d[(k),(j)] = d[(j),(k)] \quad (\textit{Complete link}).$$

*Stap 4*

Als  $k = N - 1$ , dan stop, anders naar stap 2.

De *single link* (ook wel de *Minimum Spanning Tree* – MST – genoemd) is door het gebruik van de min-operator de meest “*progressieve*” hiërarchische clustermethode, terwijl de *complete link* door het gebruik van de max-operator de meest “*behoudende*” hiërarchische clustermethode is. Een groot aantal varianten is hiertussen te bedenken.

## 11.2. Partitionele clustering

Het doel van een *partitionele clustermethode* is een gegeven dataset van  $N$  objecten in precies  $m$  disjuncte deelverzamelingen op te delen ( $m \ll N$ ), zodanig dat alle objecten die een “*natuurlijke*” samenhang vertonen zijn ondergebracht in dezelfde cluster, terwijl ongerelateerde objecten juist in verschillende clusters zijn ondergebracht. Zelfs voor bescheiden waarden voor  $m$  en  $N$ , is een *complete enumeratie* van alle mogelijke partities onwerkbaar (Stirling getal van de tweede orde): bijv.  $N = 19$  en  $m = 3$  levert  $1.93 \times 10^8$  mogelijke partities op. Twee zaken zijn dan nodig:

1. een *efficiënt zoekalgoritme*, en
2. een *criteriumfunctie* die de “*geschiktheid*” van een oplossing evalueert.

De oplossing is traditioneel geformuleerd als een *iteratieve optimalisering* van de criteriumfunctie. Als criteriumfunctie wordt veelal het kwadratisch foutcriterium gehanteerd; per cluster

$$\varepsilon_k^2 = \sum_{i \in I_k} (\mathbf{x}_i - \boldsymbol{\mu}_k)^T (\mathbf{x}_i - \boldsymbol{\mu}_k)$$

en voor de gehele clustering

$$E_m^2 = \sum_{k=1}^m \varepsilon_k^2.$$

De clusterprocedure is er dus op gericht  $E_m^2$  te minimaliseren. De meeste implementaties vereisen een *initiële clustering* (random gekozen of gebaseerd op a priori kennis); de hieruit berekende *clustercentra* vormen de basis voor een *herclassificatie-stap* waarbij ieder object wordt toegewezen aan het dichtstbijliggende clustercentrum. Hierna volgt een berekening van nieuwe clustercentra, waarna wederom herclassificatie plaats vindt; enzovoorts.

Het klassieke algoritme (FORGY) bestaat dan uit de volgende stappen:

*Stap 1*

Initialiseer  $m$  clustercentra (meestal een random keuze uit de  $N$  objecten).

*Stap 2*

Wijs clusterlabels toe aan alle objecten op basis van de kleinste afstand tot een der clustercentra.

*Stap 3*

Bereken nieuwe clustercentra.

*Stap 4*

Alterneer stap 2 en stap 3 totdat geen veranderingen meer optreden (convergentie) of tot een van te voren ingesteld aantal iteraties is bereikt.

ISODATA is een van de meest populaire implementaties die aan FORGY nog de volgende stappen toevoegt:

*Stap 5*

Verwijder clusters met minder dan  $k$  objecten en behandel deze objecten als uitbijters (outliers).

*Stap 6*

Breek clusters op of voeg clusters samen (parameters door de gebruiker te specificeren).

*Stap 7*

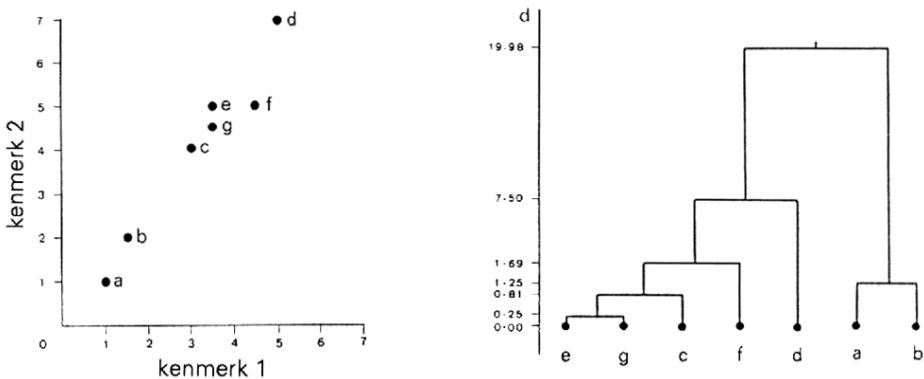
Bereken nieuwe clustercentra en herhaal stap 2 tot en met stap 6.

In het algemeen staan de implementaties het gebruik van zowel de Euclidische afstand als de Mahalanobis afstand toe.

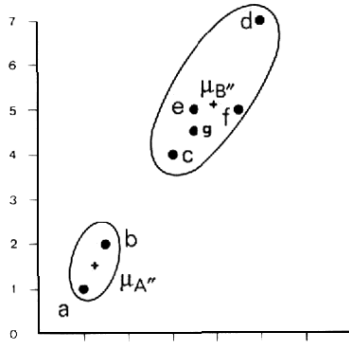
Op basis van de reeds eerder besproken concepten van het schatten van kansdichtheden kennen wij de clusterprocedures “*Mode-seeking*”-algoritmen. Met behulp van een hypersfeer of hyperkubus kan een lokale schatting plaatsvinden van de dichtheid. Hierbij is de dichtheid in een punt in de objectruimte proportioneel met het aantal objecten in de omgeving van dat punt. De procedure voorziet dan vervolgens in een *heuvelklim*-benadering om de aanwezige *modes* te detecteren. Objecten worden dan ingedeeld op basis van de dichtstbijzijnde mode (clustercentrum).

Naast de vele alternatieve procedures vormen de iteratieve procedures gebaseerd op *fuzzy sets* een interessante categorie. Hierbij wordt het unieke clusterlabel (ieder object behoort slechts tot één cluster) vervangen door de *clusterlidmaatschapswaarde* (ieder object behoort met een zekere lidmaatschapswaarde tot iedere cluster). In zekere zin verkrijgt men door de lidmaatschapswaarden additionele informatie over de resulterende clustering. Een maat voor de validiteit van de clustering is daarmee eenvoudig te realiseren.

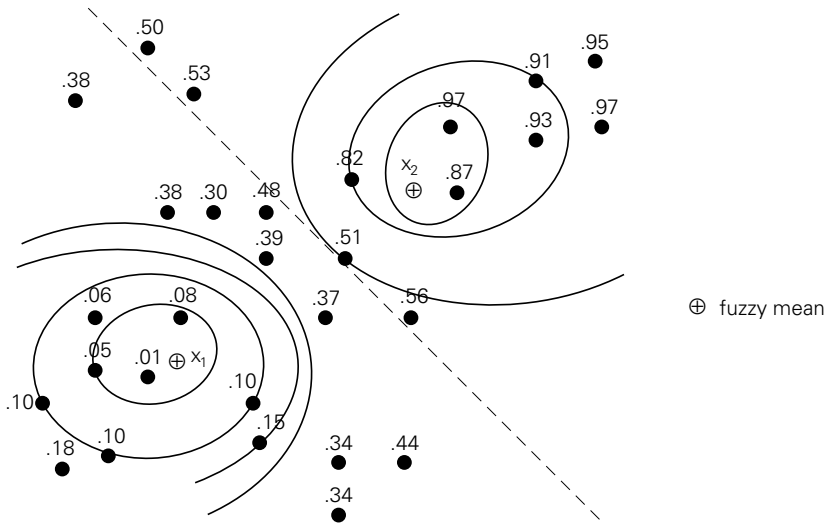
In de figuren 11.3, 11.4 en 11.5 zijn voorbeelden gegeven van resultaten van respectievelijk een hiërarchische procedure, een partitionele procedure en een fuzzy partitionele procedure.



**Figuur 11.3.** Voorbeeld oplossing van een hiërarchische clusterprocedure.



**Figuur 11.4.** Voorbeeld oplossing van een partionele clusterprocedure.



**Figuur 11.5.** Voorbeeld van een fuzzy-partionele clusterprocedure; lidmaatschapswaarden tot één van de clusters.

# 12

## Fuzzy sets

In het voorgaande is op twee plaatsen (hoofdstuk 9 en 11) reeds betoogd dat er gevallen denkbaar zijn dat “*harde*” labels (een object behoort wel of niet tot een objectklasse of gedetecteerde cluster) een te absolute uitspraak is voor sommige objecten of voor sommige classificatietaken. Ook is gesteld dat de statistische onderbouw voor classificatietaken niet altijd bruikbaar is en dat in die gevallen teruggevallen moet worden op de *verzamelingeleer* (*possibilistisch* in plaats van *probabilistisch*).

Het betrekken van de theorie der vage verzamelingen (fuzzy sets) bij de patroonherkenning brengt echter een aantal complicaties met zich mee die op zijn minst enig commentaar verdienen.

In veel gevallen is patroonherkenning op te vatten als een hybride probleem:

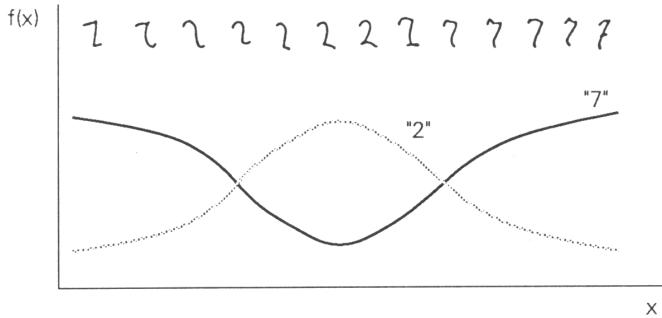
**objectwaarneming** = objectprototype + *ruis*.

De interactie tussen *prototype* en *ruis* is het best aan te pakken met *statistische* benaderingen. Is het *structurele verband* in de prototypen dominant ten opzichte van de ruis dan is een *syntactische* benadering (gericht op de intrinsieke structuur) voor de hand liggend. Bestaat er echter *vaagheid* over wat als prototype van een objectklasse opgevat kan worden dan ligt het voor de hand deze intrinsieke vaagheid te modelleren met behulp van daartoe geëigend gereedschap (*fuzzy set theorie*). Wij spreken dan van “*vage klassen*”: verzamelingen objecten waarvoor geen precieze criteria te geven zijn voor het al of niet behoren tot die klassen.

---

### Notaties hoofdstuk 12

$\chi(\cdot)$	karakteristieke functie	$\tilde{A} \oplus \tilde{B}$	symmetrisch verschil tussen twee vage verzamelingen
$f(\cdot), g(\cdot)$	lidmaatschapsfuncties	$F(\delta)$	nivea verzameling als functie van de niveauparameter $\delta$
$\tilde{A}, \tilde{B}$	vage verzamelingen	$I(f)$	hoeveelheid vaagheid in $f$
$\tilde{A} \cap \tilde{B}$	doorsnede van twee vage verzamelingen	$v(\cdot, \cdot)$	vage relatie
$\tilde{A} \cup \tilde{B}$	vereniging van twee vage verzamelingen	$\mathcal{B}$	vage objectbeschrijving
$\neg \tilde{A}$	complement van vage verzameling $\tilde{A}$	$\mathcal{D}$	afstandscriterium voor selectie van vage kenmerken
$f_{\tilde{A}}(\mathbf{x})$	mate van lidmaatschap van $\mathbf{x}$ tot de vage verzameling $\tilde{A}$		



**Figuur 12.1.** Voorbeeld van intrinsieke “vage” klassen: een vage klasse “2” en een vage klasse “7”.

In figuur 12.1 is een situatie geschetst waarbij het onduidelijk is waar de klasse “2” begint en eindigt en waar de klasse “7”.

In termen van de traditionele patroonherkenning staan wij dan voor tenminste twee problemen:

- n in geval van *ambigue* labels kan van *klasconditioneel leren geen sprake* zijn;
- n in geval van *ambigue* labels kan van een *evaluatie* van het classificatiesysteem met behulp van de traditionele *foutenanalyse geen sprake* zijn.

De ingrediënten van een vage-verzamelingenleer moeten hiervoor adequate oplossingen kunnen bieden. Wij zullen hiertoe een aantal *operatoren* de revue laten passeren welke ons kunnen helpen het classifierontwerp te onderbouwen.

De *lidmaatschapsfunctie*  $f(\cdot)$ , gedefinieerd op  $[0,1]$ , is een generalisatie van de karakteristieke functie  $\chi$  op  $\{0,1\}$ .

Een aantal elementaire operaties is een directe generalisatie van de operaties op karakteristieke functies in de traditionele verzamelingenleer. De *doorsnede* van twee vage verzamelingen  $\tilde{A}$  en  $\tilde{B}$  is dan gegeven door:

$$f_{\tilde{A} \cap \tilde{B}}(\mathbf{x}) = \text{Min}[f_{\tilde{A}}(\mathbf{x}), f_{\tilde{B}}(\mathbf{x})],$$

de *vereniging* door:

$$f_{\tilde{A} \cup \tilde{B}}(\mathbf{x}) = \text{Max}[f_{\tilde{A}}(\mathbf{x}), f_{\tilde{B}}(\mathbf{x})]$$

en het *complement* door:

$$f_{\neg \tilde{A}}(\mathbf{x}) = 1 - f_{\tilde{A}}(\mathbf{x}).$$

In figuur 12.2 zijn genoemde operatoren geïllustreerd.

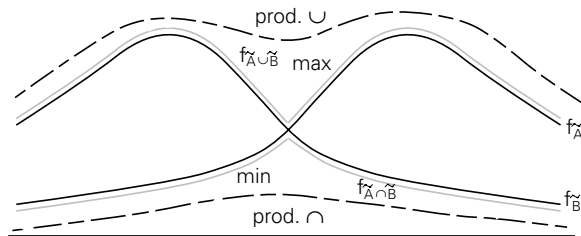
Een *zwakkere* (niet idempotente) doorsnede kan worden verkregen met:

$$f_{A \cap \tilde{B}}(\mathbf{x}) = f_{\tilde{A}}(\mathbf{x}) \cdot f_{\tilde{B}}(\mathbf{x})$$

en een *sterkere* vereniging met:

$$f_{A \cup \tilde{B}}(\mathbf{x}) = f_{\tilde{A}}(\mathbf{x}) + f_{\tilde{B}}(\mathbf{x}) - f_{\tilde{A}}(\mathbf{x}) \cdot f_{\tilde{B}}(\mathbf{x}).$$

Wij merken op dat zowel de *max*- of *min*-operator als de *produkt*-operator legitieme generalisaties vanuit de klassieke verzamelingenleer opleveren, maar dat de eigenschappen van de produkt-operator mathematisch minder aantrekkelijk zijn. Toch blijkt in de praktijk (H.J. Zimmermann, *Fuzzy Set Theory and its Applications*, Boston, 1985) dat de produkt-operator verrassend goed aansluit bij het *menselijk-subjectief* redeneren.



**Figuur 12.2.** Illustratie van de doorsnede en de vereniging van twee vage verzamelingen  $A$  en  $B$ ; (de *min*-, *max*- en *produkt*-operator).

Net zoals in het statistisch domein willen wij ook hier – zij het dan in possibilistische termen – een afstand tussen twee (vage) klassen definiëren. De “inter-fuzzy set”-afstand wordt verkregen door het “symmetrisch verschil” uit de klassieke verzamelingenleer te generaliseren. Het symmetrisch verschil is gegeven door:

$$\tilde{A} \oplus \tilde{B} = (\tilde{A} \cap \neg \tilde{B}) \cup (\neg \tilde{A} \cap \tilde{B}).$$

Toepassen van de *max*-*min*-operatoren levert

$$f_{\tilde{A} \otimes \tilde{B}}(\mathbf{x}) = \max[\min[f_{\tilde{A}}(\mathbf{x}), f_{\neg \tilde{B}}(\mathbf{x})], \min[f_{\neg \tilde{A}}(\mathbf{x}), f_{\tilde{B}}(\mathbf{x})]],$$

maar ook

$$g_{\tilde{A} \otimes \tilde{B}}(\mathbf{x}) = |f_{\tilde{A}}(\mathbf{x}) - f_{\tilde{B}}(\mathbf{x})|$$

met de eigenschap dat

$$g_{\tilde{A} \otimes \tilde{B}}(\mathbf{x}) \leq f_{\tilde{A} \otimes \tilde{B}}(\mathbf{x}) \quad \text{voor alle } \mathbf{x}.$$

De inter-fuzzy set afstand geeft – net als de probabilistische afstand – aan hoever twee vage verzamelingen uit elkaar liggen. Van grote betekenis is de zogenaamde “niveau-

verzameling” (level set). Dit is een klassieke verzameling welke door het hanteren van een niveau-parameter van een vage verzameling kan worden afgeleid:

$$F(\lambda) = \{\mathbf{x} | f(\mathbf{x}) \geq \lambda\}, \quad \lambda \in [0,1].$$

Het resultaat van een *vage classificatie* (bijv. in  $m$  vage verzamelingen) kan door de niveau-parameter  $\lambda$  worden afgebeeld op een stelsel van  $m$  klassieke verzamelingen. Hierbij fungeert  $\lambda$  als een beslissingsdrempel en noemen wij de vage classificatie gegeneraliseerd consistent indien een waarde van de vage classificatie  $\{f_1, f_2, \dots, f_m\}$  afgebeeld kan worden op een complete partitie van disjuncte klassieke verzamelingen. Hierbij geldt als randvoorwaarde dat

$$\sum_{i=1}^m f_i(\mathbf{x}) = 1 \quad \text{voor alle } \mathbf{x}.$$

Analoog aan de relatie die in het statistisch domein bestaat tussen de klasconditionele *entropie* en de *foutkans*, kunnen wij hier een *vaagheidsmaat*  $i[\cdot]$  introduceren die tot uitdrukking brengt hoe vaag een classificatie of beslissing is. Wij definiëren:

$$i[f(\mathbf{x})] = |f(\mathbf{x}) - \chi_{F(\frac{1}{2})}(\mathbf{x})|$$

waarin  $\chi_{F(\frac{1}{2})}(\mathbf{x}) = 1$  indien  $\mathbf{x} \in F(\frac{1}{2})$   
 $= 0$  indien  $\mathbf{x} \notin F(\frac{1}{2})$ .

Wij zien dat de vaagheid in  $f(\mathbf{x})$  gerelateerd is aan de bijbehorende niveauverzameling  $F(\lambda = \frac{1}{2})$ . Vaagheid is dus gerelateerd aan *beslissings-onzekerheid*. De hoeveelheid vaagheid in de vage verzameling  $f$  wordt dan uitgedrukt door

$$\begin{aligned} I(f) &= \frac{1}{N} \sum_{\mathbf{x}} i[f(\mathbf{x})] = \\ &= \frac{1}{N} \sum_{\mathbf{x}} |f(\mathbf{x}) - \chi_{F(\frac{1}{2})}(\mathbf{x})|. \end{aligned}$$

Met behulp van de niveauparameter en de hoeveelheid vaagheid in een fuzzy partitie kunnen wij het resultaat van een fuzzy classificatie redelijk evalueren. Tenslotte moeten wij nog de “*vage relatie*” introduceren:

$$v: X \times X \rightarrow [0,1].$$

Ook voor  $v$  gelden analoog de definities voor complement, doorsnede en vereniging. De vage relatie die tussen object  $\mathbf{x}$  en object  $\mathbf{y}$  bestaat is gegeven door  $v(\mathbf{x}, \mathbf{y})$  waarvoor kan gelden:

symmetrie:  $v(\mathbf{x}, \mathbf{y}) = v(\mathbf{y}, \mathbf{x})$  voor alle  $\mathbf{x}$  en  $\mathbf{y}$

(anti-)reflexiviteit:  $v(\mathbf{x}, \mathbf{y}) = (0) 1$  voor  $\mathbf{x} = \mathbf{y}$

transitiviteit:  $v(\mathbf{x}, \mathbf{z}) > \max[\min[v(\mathbf{x}, \mathbf{y}), v(\mathbf{y}, \mathbf{z})]]$ .

Gelijk de euclidische afstand tussen twee objecten in de objectruimte een zeer belangrijke (numerieke) rol speelde in de statistische benadering, zo speelt de vage relatie tussen twee objecten een dominerende rol bij een fuzzy classifier.

### 12.1. Een fuzzy classificatieregel

Wij gaan uit van de volgende a priori kennis (de leerverzameling):

object	beschrijving	geassocieerd met $m$ klasselabels
$\mathbf{x}_1$	$\mathcal{B}_1$	$f_1(\mathbf{x}_1) f_2(\mathbf{x}_1) \dots f_m(\mathbf{x}_1)$
$\mathbf{x}_2$	$\mathcal{B}_2$	$f_1(\mathbf{x}_2) f_2(\mathbf{x}_2) \dots f_m(\mathbf{x}_2)$
$\vdots$	$\vdots$	$\vdots$
$\mathbf{x}_N$	$\mathcal{B}_N$	$f_1(\mathbf{x}_N) f_2(\mathbf{x}_N) \dots f_m(\mathbf{x}_N)$

Een onbekend object  $\mathbf{x}'$  met een beschrijving  $\mathcal{B}'$  wordt nu op de volgende wijze geclassificeerd.

Beschouw de relatie

$$v(\mathbf{x}_i, \mathbf{x}') = \{ \text{de mate waarin beschrijving } \mathcal{B}_i \text{ en } \mathcal{B}' \text{ op elkaar lijken} \}$$

zo ook  $v(\mathbf{x}_1, \mathbf{x}'), v(\mathbf{x}_2, \mathbf{x}'), \dots, v(\mathbf{x}_N, \mathbf{x}')$ .

Dan is vervolgens de lidmaatschapswaarde van  $\mathbf{x}'$  tot de  $j$ -de vage klasse gegeven door het maximum te nemen over alle  $\mathbf{x}$  uit de leerverzameling van de compositie van de relaties  $v(\mathbf{x}, \mathbf{x}')$  en  $v(\mathbf{x}, j\text{-de vage klasse}) = f_j(\mathbf{x})$ , dus:

$$f_j(\mathbf{x}') = \max[\min[v(\mathbf{x}, \mathbf{x}'), f_j(\mathbf{x})]].$$

**NB.** De compositie van twee vage relaties is hier gedefinieerd als de doorsnede van beide relaties.

In de context van vage classificaties zullen de objecten veelal evenzo vaag beschreven zijn. Wij beschouwen de verzameling van vage uitspraken over  $k$  mogelijke eigenschappen (kenmerken), waarvan ieder gesteld kan zijn in bijvoorbeeld de volgende termen:

de *i*-de eigenschap is “waar”  
 “min of meer waar”  
 “grensgeval”  
 “min of meer onwaar”  
 “onwaar”

Voor deze *i*-de eigenschap is dan de relatie  $v_i(\mathbf{x}, \mathbf{x}')$  gegeven door:

$$v_i(\mathbf{x}, \mathbf{x}') = \max[\min[f_{\text{waar}}(\mathbf{x})_i, f_{\text{waar}}(\mathbf{x}')_i],$$

.....

$$\min[f_{\text{grens}}(\mathbf{x})_i, f_{\text{grens}}(\mathbf{x}')_i],$$

.....

$$\min[f_{\text{onwaar}}(\mathbf{x})_i, f_{\text{onwaar}}(\mathbf{x}')_i]]$$

Door de relatie tussen twee objecten niet sterker te definiëren dan de zwakste eigenschapsovereenkomst verkrijgen wij:

$$v(\mathbf{x}, \mathbf{x}') = \min_i [v_i(\mathbf{x}, \mathbf{x}')].$$

De hiervoor gedefinieerde fuzzy-classificator acteert derhalve als volgt:

De relatie (en dus de uiteindelijke fuzzy-labelling) tussen een onbekend object  $\mathbf{x}'$  en de vaag gelabelde leerverzameling wordt verkregen door te maximaliseren over de gehele leerverzameling (naaste nabuur benadering) van de object-object-relaties waarbij iedere object-object-relatie niet sterker is dan de zwakste eigenschapsovereenkomst tussen die objecten.

Tot slot vragen wij ons nog af welke eigenschappen (kenmerken) het meest geschikt zijn voor een fuzzy classificator.

## 12.2. Selectie van geschikte vage kenmerken

Ook bij de fuzzy classificatie – zo stelden wij reeds – doet zich de vraag voor in welke mate vage kenmerken of kenmerkdeelverzamelingen bijdragen tot de uiteindelijke classificatie (met uiteraard de minste hoeveelheid resulterende vaagheid). Ook dit is weer te stellen als een optimalisatie-probleem met een maat voor vaagheid (betrokken op de vage leerverzameling) als criteriumfunctie. De fundamentele problematiek is identiek aan die welke wij in het statistisch domein tegen kwamen, namelijk:

1. bestaat er een monotoon evaluatiecriterium voor kenmerkdeelverzamelingen,
2. bestaat er een efficiënt zoekalgoritme ter voorkoming van de complete enumeratie.

Beschouwen wij de kenmerkdeelverzameling  $\mathcal{F}_k$  bestaande uit  $k$  vage kenmerken. Dan voldoet de volgende evaluatiefunctie:

$$\mathcal{D} = \sum_{i < j} (\max_k \sum_{\substack{1 \\ 2 \\ 3}} [|\mathcal{B}_{ik} - \mathcal{B}_{jk}|] \sum_{\substack{1 \\ 2 \\ 3}} \sum_m |\lambda_{im} - \lambda_{jm}|)$$

waarin 1, 2, 3 slaan op respectievelijk *waar*, *grensgeval* en *onwaar*. In het vage-kenmerkdomein wordt als functie van  $k$  het maximale verschil in de beschrijvingen  $\mathcal{B}$  gezocht bij een gegeven label-discrepanantie in het vage-labeldomein.  $\mathcal{D}$  is een monotoon criterium en staat de implementatie van een efficiënt zoekalgoritme gemakkelijk toe (bijvoorbeeld het eerder besproken Branch and Bound algoritme).

# 13

## Kennisgestuurde systemen

In de voorgaande hoofdstukken hebben wij de patroonherkenning gemodelleerd volgens een aantal rechtlijnige principes:

1. Aan een object kun je enige metingen verrichten (*kenmerkbe­paling*) (= *abstractie*)
2. Een verzameling objecten met bijbehorende interpretatie (*klasselabels*) vormt de leerverzameling op basis waarvan een gegeneraliseerde klasse­beschrijving kan worden afgeleid (= *generalisatie*)
3. Een beslissingsregel kan op basis van én de objectabstractie én de gegeneraliseerde klasse­beschrijvingen een uitspraak doen over ieder nieuw aan te bieden object (= *classificatie*).

Ogenschijnlijk vertoont dit proces enige analogie met het menselijk herkenning­proces: bij het waarnemen van een object (patroon) – bijvoorbeeld door kijken – (de uiteindelijke abstractie is onbekend) raadpleegt de waarnemer zijn geheugen of hij zoiets al ooit eens eerder heeft gezien (raadplegen van gegeneraliseerde klasse­beschrijvingen). Na korte of langere tijd kan dan de waarnemer tot de conclusie komen dat hij óf soortgelijke objecten nog nooit heeft waargenomen, óf dat hij met een zekere waarschijnlijkheid meent dat het object geïnterpreteerd moet worden als zijnde een mogelijke representant van een bepaalde klasse (classificatie). Een voorbeeld is het “lezen” van – met de hand geschreven – karakters. Het is niet duidelijk hoe en op welk niveau objectabstractie bij de mens plaats vindt, noch hoe de klassegeneralisaties zijn gerepresenteerd. Wel is duidelijk dat veel andere aspecten de uiteindelijke beslissing(en) kunnen beïnvloeden:

- details van het handschrift (zelfs een persoonsidentificatie is mogelijk),
- tekstuele context (semantiek),
- gebruikte taal, grammaticale regels,

---

### Notaties hoofdstuk 13

$\mathcal{H}_i$	$i$ -de hypothese	$\mathcal{M}b$	measure of belief
H	verzameling van alle hypothesen	$\mathcal{M}d$	measure of disbelief
$b_j$	$j$ -de bewijslast	$\varphi$	basic probability
$Cf$	certainty factor	$\wedge$	doorsnede-operator
$\mathcal{F}$	support function	$\text{bel}(\cdot)$	belief function

- (aard van eventueel cijfermateriaal, geldbedragen bij uitverkoop zijn anders dan lottocijfers),
- onderwerp en bedoeling van de tekst,
- tijd, plaats en omgeving waarin het werd geschreven,
- de aard van de informatiedrager,
- de mate van zekerheid/twijfel over voorgaande letter- cq. woordclassificaties,
- etc. etc.

Eén ding is duidelijk: hoe de mens deze “*kennis*” aanwendt is meestal niet eenvoudig te expliciteren, maar feit is dat de mens mag worden beschouwd als een “*expert*”-karakterherkenner.

Zulk “*expert*”-gedrag doet zich veelvuldig voor, bijvoorbeeld in de medische diagnostiek, de luchtfoto-interpretatie, de gewasclassificatie, de wolkeninterpretatie en de weersvoorspelling, enzovoort.

In al die gevallen zullen rechtlijnige procedures, zoals beschreven, maar zeer beperkt presteren. Het is in dat licht dat in het laatste decenium nadrukkelijk een streven bestaat om herkenningprocedures te omgeven met zogenaamde *kennissystemen*. Met de hierin ondergebrachte additionele kennis, heuristische kennis (vuistregels) en dergelijke kan worden gemanipuleerd en de feitelijke herkenning worden gestuurd. Wij spreken van *gestructureerde kennisintegratie* waarbij kennis wordt onttrokken aan het specifieke toepassingsgebied en vervolgens gecodeerd in een manipuleerbare vorm (bijvoorbeeld *regels*). Aldus richt men zich op het ontwerpen van flexibele systemen in termen van een voortdurende wisselwerking tussen verwerkingsmethode (herkenningmethode), aanwezige kennis en aangeboden data.

Tegenover het *conventionele* computerprogramma waarin relevante kennis en methoden om deze kennis te gebruiken sterk verweven en verstrengeld zijn (en daardoor slecht te wijzigen), staat het *kennisprogramma* waarin een stricte scheiding is aangebracht tussen algemene kennis over het werkelijke probleem (→ *kennisbestand*), de informatie over werkelijke data (*input data*) en de methode om de algemene kennis toe te passen (→ *inferentiemechanisme*). Hierdoor is zowel wijzigen als aanvullen relatief eenvoudig. Het vereist echter een efficiënte, manipuleerbare vorm van kennisrepresentatie.

Een systeem voor kennisgestuurde patroonherkenning zal de volgende karakteristieken hebben:

- krachtig: een intelligent gedrag in een complexe herkenningstaak,
- robuust: geschikt om met incomplete, vage data te handelen,
- flexibel: geschikt om met incomplete, vage kennis te handelen,
- opportunistisch: in staat om kennis aan te wenden op het “juiste” moment,
- transparant: in staat om de redenering te verklaren en beslissingen te onderbouwen.

### 13.1. Kennis en kennissystemen

In de hiervoor gegeven karakterisering van een systeem voor kennisgestuurde patroonherkenning werd formeel onderscheid gemaakt tussen kennis en data. Kennis beschrijft de relaties tussen gegeneraliseerde verzamelingen van objecten of situaties, terwijl data de feiten geassocieerd met een specifiek object of situatie beschrijft. Naast algemene kennis over het probleem is met name “ervaringskennis” van belang. Ervaringskennis representeert “hands-on”-ervaring en paart feiten aan heuristieken. Een heuristiek is een vuistregel of ander mechanisme of vereenvoudiging welke het zoeken reduceert of beperkt in grote oplossingsruimten. In tegenstelling tot een algoritme garandeert een heuristiek geen correcte oplossing.

In het bijzonder geldt voor de ervaringskennis het probleem van de *geldigheid* (zeker of onzeker). Op dit probleem komen wij nog kort terug. Als gezegd, het representeren van kennis in een manipuleerbare vorm is voor het ontwerpen van het systeem van groot belang. Twee concepten worden vaak tesamen in een systeem gebruikt, namelijk het *frame-concept* en het *regel-concept*.

Het frame-concept is een kennisrepresentatieschema dat aan een object of situatie een verzameling eigenschappen toekent (feiten, regels, defaults, actieve waarden), tesamen met relaties. (wij spreken ook wel van een eigenschappenlijst of record; omdat frames naar elkaar verwijzen spreken wij ook van *semantische netwerken*).

Het regel-concept is een conditionele bewering bestaande uit twee delen:

- de IF clause, de conditie waaraan voldaan moet zijn,
- de THEN clause, de actie welke ondernomen moet worden (wij spreken ook wel van situatie-actie regel of produktie).

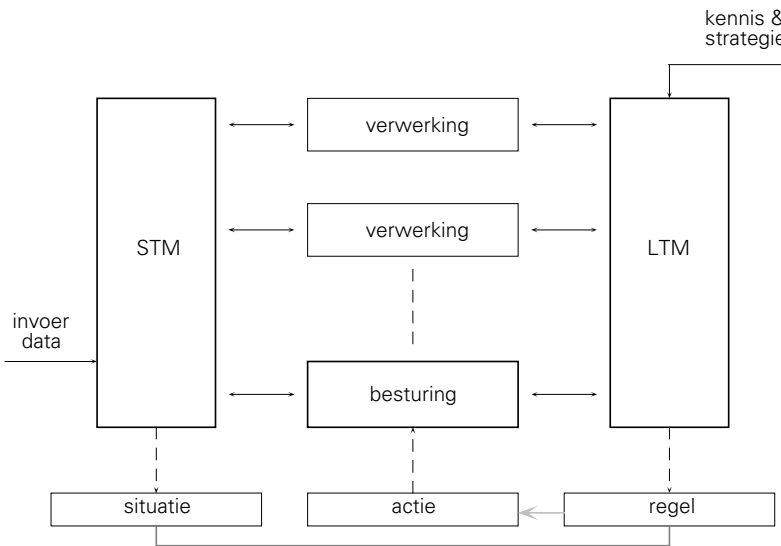
Zogenaamde “rule-based”-systemen zijn geheel gedefinieerd in termen van situatie-actie regels. Een kennissysteem van dit type is gekenmerkt door de volgende rede-nertrant:

- pas een regel toe
- bekijk het resultaat
- pas een nieuwe regel toe op de (mogelijk) gewijzigde situatie
- .....
- enzovoort.

Wij maken onderscheid tussen *voorwaarts redeneren* (*inferentie*) en *achterwaarts redeneren* (*deductie*). Bij inferentie wordt gewerkt vanuit de initiële gegevens en door toepassen van regels getracht de oplossing te bereiken. Bij deductie tracht men een mogelijke oplossing door toepassen van regels te verifiëren door naar de bestaande gegevens toe te werken.

Qua datastructuur is het model voor kennisgestuurde systemen voor patroonherkenning in de meeste gevallen terug te voeren tot het zogenaamde “black-board” model.

Hierin communiceren de op zich kennisvrije procedures met een databestand (het blackboard) waarin ondermeer kennis expliciet is ondergebracht. Voortbordurend op dit model heeft men vervolgens een geheugenpartitie aangebracht: een *werkgeheugen* (het korte termijn geheugen) en een *kennisgeheugen* (het lange termijn geheugen). In het werkgeheugen vinden wij de inputdata, kenmerk- en parameterwaarden, uitvoerresultaten en dergelijke; het lange termijn geheugen bevat de produktieregels voor *verwerking*, *besturing* en *strategie*, alsmede objectframes met hun onderlinge relaties. Alle modules hebben toegang tot beide geheugenblokken. In figuur 13.1 is een en ander schematisch weergegeven.



**Figuur 13.1.** Schematische weergave van het "black-board"-systeem en de rol van situatie-actieregels.

Een situatie, aangetroffen in het werkgeheugen, doet een regel afvuren welke het vervolg van het proces bepaalt. Naast de situatie-actie regels waarin het bepalen van kenmerkwaarden en de wijze van verwerken wordt geregeld, is een belangrijk deel van de beschikbare kennis neergelegd in *besturingsregels* en *strategieregels*. In de besturingsregels wordt de ordening van de te gebruiken situatie-actie regels aangegeven, te volgen acties gespecificeerd, evaluatie aangeroepen en het eventueel stoppen voorbereid. De strategie regels – op basis van aangeboden data en evaluatieparameters – kiezen de besturingsregels en beslissen dynamisch over de te volgen weg. Samenvattend is de besturingsstrategie gericht op:

- de volgorde waarin de verschillende heuristieken moeten worden toegepast en
- het pad waarlangs op de objecten de heuristieken worden getest  
(dus: een dynamische ordening van regels en een dynamische keuze van het verwerkingspad).

In veel gevallen is de situatie gekenschetst door een conditie, bijvoorbeeld uitgedrukt door:

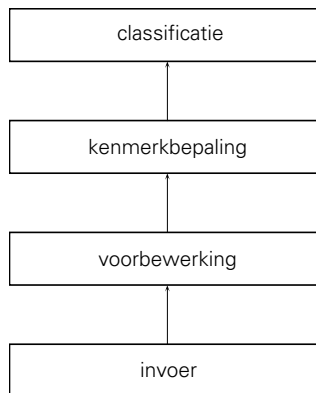
- zeer hoog      *waar*
- hoog            *min of meer waar*
- medium        *grensgeval*
- laag            *min of meer onwaar*
- zeer laag      *onwaar.*

en de actie gericht op modificatie van classificatietask, kenmerkbeplating en dergelijke.

### 13.2. Besturingsmechanismen in de patroonherkenning

De taak van het besturingsmechanisme – zo zagen wij – is het tot uitvoer brengen en evalueren van potentiële acties. Ook zagen wij dat er twee essentieel verschillende strategieën (redeneerwegen) bestonden:

- inferentie      “bottom-up”      *data gestuurd,*
- deductie        “top-down”        *model gestuurd.*



**Figuur 13.2.** *Traditioneel patroonherkenningsproces: “bottom-up”-analyse.*

In figuur 13.2 is het typische patroonherkenningsproces weergegeven. Na een nauwkeurige “preprocessing” worden de kenmerkwaarden bepaald op grond waarvan de uiteindelijke classificatie tot stand komt. De “bottom-up” benadering gaat van de ingevoerde data uit (data driven) en tracht via een groot aantal stappen tot interpretatie van het object te komen. De strategie is alleen acceptabel indien de invoerdata nauwkeurig en betrouwbaar is en geen ambigue informatie oplevert voor hoger-niveau processen. In praktische problemen is daaraan maar zelden volledig voldaan. Bij “top-down” besturing start men af met een *hypothes*e over het betreffende object zonder directe referentie met de invoerdata. Deze hypothes e bestaat weer uit sub-hypothesen waaraan voldaan moet zijn wil de oorspronkelijke hypothes e waar zijn. Zo gaat het proces voort totdat de sub-...-sub-hypothesen eenvoudig genoeg zijn om direct

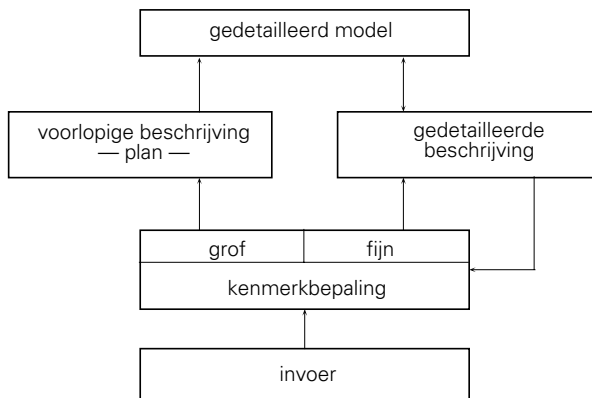
geverifieerd te kunnen worden uit de invoerdata. Omdat deze besturing uitgaat van een model en vervolgens tracht te bewijzen dat het model overeenkomt met de werkelijke invoerdata wordt deze besturing “model-driven” genoemd. Als de keuze van het pad waarlangs achtereenvolgens sub-hypothesen worden getoetst een “juist” of “geschikt gekozen” pad is, dan is “top-down” analyse snel en voorkomt veel laag-niveau rekenwerk omdat men gericht bezig is. Is de keuze van de hypothese “slecht”, dan zal het proces gekenmerkt zijn door overtollig zoeken en proberen.

Patroonherkenning is een typisch voorbeeld waarbij een combinatie van beide besturingen gewenst is; een besturing waarbij de voordelen van “bottom-up” en “top-down” aan elkaar gepaard zijn. Figuur 13.3 toont een gecombineerde “bottom-up” en “top-down” analyse. De achterliggende gedachte is ongeveer als volgt:

Eerst worden vanuit de ingevoerde objectdata – in grove vorm, zonder veel kennis vooraf – enige fundamentele kenmerkwaarden bepaald en wordt een ruwe – indicatieve – classificatie afgeleid; aan de hand hiervan wordt met behulp van modellen een start-hypothese en een onderzoeksplan opgesteld.

De “bottom-up” analyse initialiseert in feit een “top-down” analyse. Het oeverloos zoeken en proberen wordt omzeild en tijdvergende kenmerkextractie alleen uitgevoerd daar waar er echt reden voor bestaat.

In de laatste paragraaf, tenslotte, komen wij terug op een probleem dat bekend staat als “*inexact redeneren*”. Ongeacht de besturing willen wij weten hoe “onzekerheid” *propageert* tijdens het redeneren.



**Figuur 13.3.** Een combinatie van “bottom-up”- en “top-down”-analyse.

### 13.3. Inexact redeneren: onzekerheidscalculi

Bij het hanteren van ervaringskennis – zo zagen wij – kunnen wij er niet altijd van uitgaan dat een zekere hypothese ( $\mathcal{H}$ ) waar is als voldaan is aan bepaalde voorwaarden of sub-hypothesen (bewijslasten  $b_1, b_2, \dots$ ). Vaak is aan sommige sub-hypothesen niet

volledig voldaan of is de geldigheid betrekkelijk (uitgedrukt in termen “soms”, “waarschijnlijk”, met een bepaalde “zekerheid”). Hiertoe zijn zogenoemde “zekerheidsfactoren” (*certainty factors Cf*) geïntroduceerd met een waardebereik  $[0,1000]$ ,  $[0,1]$ , of  $[-1,1]$  met het doel aan te geven in welke mate een uitspraak waar danwel onwaar is.

Het lag voor de hand de zekerheidsfactoren in de eerste plaats te koppelen aan het kansbegrip, ook al bleek snel dat het twijfelachtig was dat een menselijke expert zich bij het redeneren en concluderen laat leiden door een kansformalisme.

Beschouwen wij de kennisregel:

$$\text{IF A en B THEN C } (p_C = 0.9). \quad (13.1)$$

Is nu  $p_C = .4$  en  $p_B = 0.9$  dan zou C geconcludeerd kunnen worden met kans  $\hat{p} = 0.36$ . Deze uitkomst stoelt op een rekenregel als:

$$\hat{p}_C = p_C \min[p_A, p_B].$$

Luidt de kennisregel echter:

$$\text{IF E of D THEN C } (p_C = 0.6), \quad (13.2)$$

dan volgt, indien  $p_E = 0.3$  en  $p_D = 0.5$ , dat C geconcludeerd kan worden met  $\hat{p}_C = 0.3$  volgens

$$\hat{p}_C = p_C \max[p_E, p_D].$$

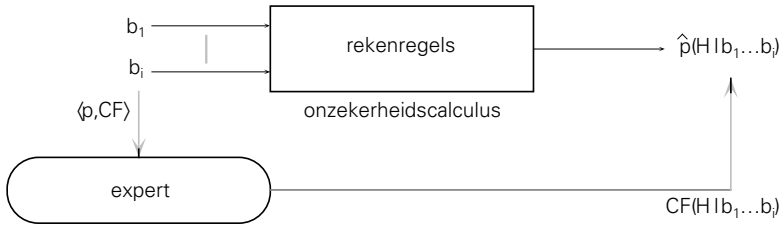
Combinatie van beide regels (ze concluderen beide immers C) kan leiden tot:

$$\text{regel (1)} \rightarrow \hat{p}_C^1 = 0.36$$

$$\text{regel (2)} \rightarrow \hat{p}_C^2 = 0.3$$

$$\hat{p}_C = \mathcal{F}(\hat{p}_C^1, \hat{p}_C^2) = 0.5$$

Er is sprake van “wederzijdse ondersteuning” en beide regels tesamen leiden tot een grotere zekerheid omtrent conclusie C. De functie  $\mathcal{F}$  modelleert in feite een stukje “menselijk redeneren”. Dit is geformaliseerd in een zogenoemde *onzekerheids calculus*: een stelsel rekenregels welke zekerheidsfactoren opwaardeert en toekent aan de uiteindelijke conclusies (op grond van meervoudige bewijsgegevens). Figuur 13.4 geeft daarvan een schematische voorstelling.



**Figuur 13.4.** Schematische voorstelling van de rol van een onzekerheids calculus.

In de figuur is tevens de mogelijke – en in veel gevallen waarschijnlijke – discrepantie aangegeven welke kan bestaan tussen de “ervaringswereld” van de expert  $\langle Cf \rangle$  en de op rekenregels gebaseerde modelmatige “zekerheid”  $\langle p \rangle$ . Voor iedere toepassing is het nog maar de vraag in hoeverre  $\hat{p}(\mathcal{H}|b_1, b_2, \dots)$  aansluit bij  $Cf(\mathcal{H}|b_1, b_2, \dots)$  van de expert.

Heel in het bijzonder wordt door de expert zelden de ontkenning van een hypothese ( $\neg\mathcal{H}$ ) een zekerheid  $Cf(\neg\mathcal{H}|b_1, b_2, \dots)$  meegegeven welke het complement is van  $Cf(\mathcal{H}|b_1, b_2, \dots)$ , dus

$$Cf(\neg\mathcal{H}|b_1, b_2, \dots) \neq 1 - Cf(\mathcal{H}|b_1, b_2, \dots) \text{ voor } Cf \in [0,1].$$

Op grond daarvan is een strict kansmodel waarin per definitie geldt:

$$p(\neg\mathcal{H}|b_1, b_2, \dots) = 1 - p(\mathcal{H}|b_1, b_2, \dots),$$

in het algemeen geen juiste afspiegeling van menselijke interpretatie van zekerheid.

Van de modellen die ontwikkeld zijn om met name deze stricte kanstheorie te relaxeren zijn de volgende twee het meest bekend:

- n Shortliffe-Buchanan model en
- n Dempster-Shafer model.

Het Shortliffe-Buchanan model is minder restrictief dan het probabilistische model (Bayes) en sluit beter aan bij onze menselijke intuïtie. Wij zullen hier de beide modellen naast het Bayes model kort de revue laten passeren.

#### Bayes model

Centraal staat de relatie tussen de a priori kansen en de voorwaardelijke kansen (de formule van Bayes):

$$p(\mathcal{H}|b_1, b_2, \dots) = \frac{p(b_1, b_2, \dots|\mathcal{H}) \cdot p(\mathcal{H})}{p(b_1, b_2, \dots)}.$$

Voor alle combinaties van  $b_1, b_2, \dots$  is het schatten van  $\hat{p}(b_1, b_2, \dots|\mathcal{H})$  en  $\hat{p}(b_1, b_2, \dots)$  een praktische onmogelijkheid. Hoewel in de meeste gevallen irreëel en onverdedigbaar zijn wij gedwongen *globale onafhankelijkheid*  $p(b_i) = p(b_i|b_j)$  alsmede

voorwaardelijke onafhankelijkheid  $p(b_i|b_j, \mathcal{H}) = p(b_i|\mathcal{H})$  aan te nemen. Dan kan de formule van Bayes vereenvoudigd worden tot:

$$p(\mathcal{H}|b_1, b_2, \dots) = p(\mathcal{H}) \cdot \frac{p(b_1|\mathcal{H})}{p(b_1)} \cdot \frac{p(b_2|\mathcal{H})}{p(b_2)} \cdot \dots \cdot$$

Veronderstellen wij voorts dat de verzameling hypothesen uitputtend en onderling uitsluitend zijn, dan is een en ander te herleiden tot:

$$p(\mathcal{H}_k|b_1, b_2, \dots) = p(\mathcal{H}_k) \frac{p(b_1|\mathcal{H}_k) \cdot p(b_2|\mathcal{H}_k) \cdot \dots \cdot \dots \cdot}{\sum_i p(\mathcal{H}_i) p(b_1|\mathcal{H}_i) p(b_2|\mathcal{H}_i) \cdot \dots \cdot \dots \cdot}$$

met onverminderd de conditie:

$$p(\mathcal{H}_k|b_1, b_2, \dots) = 1 - p(\neg \mathcal{H}_k|b_1, b_2, \dots).$$

Op grond van het feit dat het complete stelsel voorwaardelijke kansen bekend moet zijn is de methode niet licht implementeerbaar en vraagt zeer veel geheugen. Het volgende model is in het bijzonder gericht op het vergemakkelijken van het gebruik van voorwaardelijke kansen.

*Shortliffe-Buchanan model*

De volgende definities zijn in dit model van belang:

$\mathcal{M}b(\mathcal{H}, b_i)$ : maat voor de *toename* van het “geloof” (belief) in hypothese  $\mathcal{H}$ , gebaseerd op bewijslast  $b_i$ ;

$$\mathcal{M}b(\mathcal{H}, b_i) = \begin{cases} 1 & \text{(als } p(\mathcal{H}) = 1) \\ \frac{\max(p(\mathcal{H}|b_i), p(\mathcal{H})) - p(\mathcal{H})}{1 - p(\mathcal{H})} & \text{(anders).} \end{cases}$$

$\mathcal{M}d(\mathcal{H}, b_i)$ : maat voor de *toename* van het “ongeloof” (disbelief) in hypothese  $\mathcal{H}$ , gebaseerd op bewijslast  $b_i$ ;

$$\mathcal{M}d(\mathcal{H}, b_i) = \begin{cases} 0 & \text{(als } p(\mathcal{H}) = 0) \\ \frac{\min(p(\mathcal{H}|b_i), p(\mathcal{H})) - p(\mathcal{H})}{1 - p(\mathcal{H})} & \text{(anders).} \end{cases}$$

Uit bovenstaande wordt de zekerheidsfactor  $\langle Cf \rangle$  afgeleid:

$$Cf(\mathcal{H}, b_i) = Mb(\mathcal{H}, b_i) - Md(\mathcal{H}, b_i) \quad (Cf \in [-1, 1]).$$

Op grond van voorgaande definities is het eenvoudig in te zien dat hier

$$Cf(\mathcal{H}, b) + Cf(\neg\mathcal{H}, b) \leq 1 \quad (= 0, \text{ ingeval van binaire variabelen } \mathcal{H})$$

zal gelden mits  $p(\mathcal{H}|b_i) \geq p(\mathcal{H})$ .

Voor kleine  $p(\mathcal{H})$  vinden wij:

$$\begin{aligned} Cf(\mathcal{H}, b_i) &= Mb(\mathcal{H}, b_i) - Md(\mathcal{H}, b_i) = \\ &= \frac{p(\mathcal{H}|b_i) - p(\mathcal{H})}{1 - p(\mathcal{H})} \approx p(\mathcal{H}|b_i). \end{aligned}$$

Voor combinatie van bewijslast wordt ook in dit model globale en voorwaardelijke onafhankelijkheid verondersteld, zodat:

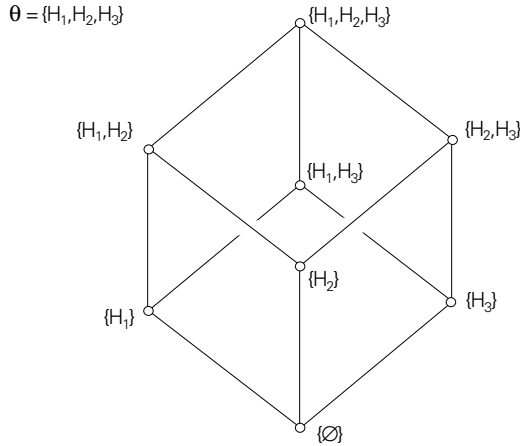
$$\begin{aligned} Cf(\mathcal{H}, b_1, b_2) &= Cf(\mathcal{H}, b_1) + Cf(\mathcal{H}, b_2) \cdot \{1 - |Cf(\mathcal{H}, b_1)|\} \\ &\quad (\text{indien } Cf(\mathcal{H}, b_1) \text{ en } Cf(\mathcal{H}, b_2) \text{ een gelijk teken bezitten}); \\ &= \frac{Cf(\mathcal{H}, b_1) + Cf(\mathcal{H}, b_2)}{1 - \min\{|Cf(\mathcal{H}, b_1)|, |Cf(\mathcal{H}, b_2)|\}} \\ &\quad (\text{indien } Cf(\mathcal{H}, b_1) \text{ en } Cf(\mathcal{H}, b_2) \text{ ongelijke tekens bezitten}). \end{aligned}$$

Wij merken op dat bij combinatie van bewijslast bijvoorbeeld  $Cf(\mathcal{H}, (b_1, b_2), b_3)$  niet noodzakelijk gelijk hoeft te zijn aan  $Cf(\mathcal{H}, b_1, (b_2, b_3))$ . Combinatie van zekerheidsfactoren is niet associatief. Dit is een zeer ongewenste situatie. Het model is zeer eenvoudig te implementeren en vraagt weinig geheugen. De opwaardering van zekerheidsfactoren (rekenregels) is heuristisch en berusten niet op een formele theorie (wat bij Bayes natuurlijk wel het geval is). De methode blijkt vrij ongevoelig te zijn voor parameter variaties.

#### *Dempster-Shafer model*

De Dempster theorie is het antwoord om formeel te ontsnappen aan de additiviteitsvoorwaarde (Bayes):  $p(\mathcal{H}) + p(\neg\mathcal{H}) = 1$ .

Dempster introduceerde zogenoemde “boven- en onder”-grenzen van de optredende relevante kansen. (Shafer verfijnde de theorie en noemde de ondergrens de “belief function” en de bovengrens de “plausibility”). De theorie is gebaseerd op stelsels verzamelingen van mogelijke hypothesen. Een volledig stelsel is gegeven in figuur 13.5.



**Figuur 13.5.** Voorbeeld van de verzameling  $\mathcal{H}$  van mogelijke deelverzameling van hypothesen bestaande uit  $\mathcal{H}_1, \mathcal{H}_2$  en  $\mathcal{H}_3$ ;  $\emptyset$  is de lege verzameling.

Het model geeft een formalisme voor het definiëren van zogenaamde “basic probabilities”, aangegeven met  $\varphi$ . Zijn er nog geen bewijslasten  $b_1, b_2, \dots$  in het spel dan geldt:

$$\varphi(\mathcal{H}) = 1 \text{ en alle overige } \varphi_i = 0 \text{ (de index } i \text{ slaat op de mogelijke deelverzamelingen van hypothesen).}$$

zodat 
$$\sum_i \varphi_i = 1$$

en 
$$\varphi(\emptyset) = 0 \text{ (per definitie).}$$

Volgt nu uit een zekere bewijslast een aanwijzing voor een bepaalde deelverzameling, zeg  $\{\mathcal{H}_1, \mathcal{H}_2\}$ , dan is:

$$\varphi(\{\mathcal{H}_1, \mathcal{H}_2\}) = \varphi_i \text{ en } \varphi(H) = 1 - \varphi_i.$$

Het combineren van bewijslast berust op het bepalen van de doorsneden van de betreffende deelverzamelingen. Het volgende voorbeeld illustreert dat.

**Voorbeeld**

Stel  $H$  bestaat uit  $\mathcal{H}_1, \mathcal{H}_2$  en  $\mathcal{H}_3$ . Op grond van bewijslast  $b_1$  wordt een “basic probability”  $\varphi_i$  toegekend aan  $\{\mathcal{H}_1, \mathcal{H}_2\}$  en  $(1 - \varphi_i)$  aan  $H$  in haar geheel. Op grond van een tweede bewijslast  $b_2$  wordt een “basic probability”  $\varphi_j$  toegewezen aan  $\{\mathcal{H}_3\}$  en  $(1 - \varphi_j)$  aan  $H$  in haar geheel. De doorsnede van  $\{\mathcal{H}_1, \mathcal{H}_2\}$  en  $\{\mathcal{H}_3\}$  is leeg (maar de lege verzameling heeft per definitie een kans gelijk aan 0). De doorsnede van  $\{\mathcal{H}_1, \mathcal{H}_2\}$  en  $H$  is  $\{\mathcal{H}_1, \mathcal{H}_2\}$  zelf, zo ook voor  $\{\mathcal{H}_3\}$  en  $H$  zelf. De basic probabilities moeten nu worden opgewaardeerd.

- Voor  $\{\mathcal{H}_1, \mathcal{H}_2\}$  wordt dat  $\varphi_i \wedge (1 - \varphi_i) / (1 - \varphi_i \wedge \varphi_i)$   
(hierin is de  $\wedge$ -operator de doorsnede-operator;  
 $\varphi_i = \varphi(\{\mathcal{H}_1, \mathcal{H}_2\} | b_1)$  en  $1 - \varphi_i = \varphi(H | b_2)$ ;  
 $(1 - \varphi_i \wedge \varphi_i)$  een normering).
- Voor  $\{\mathcal{H}_3\}$  wordt dat  $\varphi_j \wedge (1 - \varphi_j) / (1 - \varphi_j \wedge \varphi_j)$   
(hierin is  $\varphi_j = \varphi(\{\mathcal{H}_3\} | b_2)$  en  $(1 - \varphi_j) = \varphi(H | b_1)$ ).
- Voor  $H$  wordt dat  $(1 - \varphi_i) \wedge (1 - \varphi_j) / (1 - \varphi_i \wedge \varphi_j)$   
(hierin is  $(1 - \varphi_i) = \varphi(\mathcal{H} | b)$  en  $(1 - \varphi_j) = \varphi(\mathcal{H} | b)$ ).

Nadat de “basic probabilities” zijn opgewaardeerd en toegewezen kan de “*belief*” worden berekend:

$$\text{bel}(\{\mathcal{H}_1, \mathcal{H}_2\}) = \varphi(\{\mathcal{H}_1, \mathcal{H}_2\}) + \varphi(\{\mathcal{H}_1\}) + \varphi(\{\mathcal{H}_2\}).$$

De “belief” functie heeft de volgende “redelijke” eigenschappen:

$$\text{bel}(\emptyset) = 0$$

$$\text{bel}(H) = 1$$

$$\text{bel}(\mathcal{H}_i, \mathcal{H}_j) = \text{bel}(\mathcal{H}_i) + \text{bel}(\mathcal{H}_j)$$

en  $\text{bel}(\{\mathcal{H}\}) + \text{bel}(\neg\{\mathcal{H}\}) \leq 1.$

De volledige waardering wordt niet in de vorm van een getal maar in de vorm van een interval gegeven:

$$[\text{bel}(A), 1 - \text{bel}(\neg A)]$$

(hierin is de *ondergrens* de “belief” in  $A$  en de *bovengrens* de “plausibility” van  $A$ )

Het model is theoretisch goed gefundeerd; de rekenregels zijn heuristisch. Ook de bewijslasten zijn hier onafhankelijk verondersteld. De implementatie is zeer complex en vraagt veel geheugenruimte. Er is nog weinig praktijkervaring met dit model.

Inzicht in de hierboven gegeven onzekerheidscalculi en het besprokene in hoofdstuk 12 (vage verzamelingen) laten zich uitstekend combineren. De zetting van de theorie der vage verzamelingen en de problematiek van rekenregels om met onzekerheid te manipuleren vertonen grote overeenkomsten. Dempster, Dubois en Prade hebben “fuzzy” generalisaties gegeven van alle hiervoor beschreven modellen.