Eindhoven University of Technology

MASTER

EXPLORE : een methode voor het inductief leren van optimale diagnostische beslisregels

Hoffmann, Aswin L.

Award date:
1996

Link to publication

Disclaimer
This document contains a student thesis (bachelor's or master's), as authored by a student at Eindhoven University of Technology. Student theses are made available in the TU/e repository upon obtaining the required degree. The grade received is not published on the document as presented in the repository. The required complexity or quality of research of student theses may vary by program, and the required minimum study period may vary in duration.

General rights
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain

4736

TECHNISCHE UNIVERSITEIT EINDHOVEN
FACULTEIT DER ELEKTROTECHNIEK
VAKGROEP MEET- EN BESTURINGSSYSTEMEN
SECTIE MEDISCHE ELEKTROTECHNIEK

EXPLORE : EEN METHODE VOOR HET INDUCTIEF LEREN VAN OPTIMALE DIAGNOSTISCHE BESLISREGELS

A.L. HOFFMANN

september 1996

Rapport van het afstudeerwerk,
uitgevoerd van maart tot november 1994
bij de vakgroep Medische Informatica
van de Erasmus Universiteit Rotterdam,
onder verantwoording van prof.dr.ir. A. Hasman (RL/TUE)
en begeleiding van dr.ir. J.A. Kors (EUR)

DE FACULTEIT DER ELEKTROTECHNIEK VAN DE TECHNISCHE UNIVERSITEIT EINDHOVEN AANVAARDT GEEN AANSPRAKELIJKHEID VOOR DE INHOUD VAN STAGE- EN AFSTUDEERVERSLAGEN

"...Door deductie verworven wetenschappelijke kennis bestaat niet,
tenzij men de primaire premissen kent...
We moeten de primaire premissen door inductie te weten zien te komen;
want de methode volgens welke zelfs zintuiglijke waarneming
het universele implanteert, is inductief..."

Aristoteles.

Voorwoord

Omdat de eerste pagina's van een afstudeerrapport meestal het eerst (en soms alleen) worden gelezen, is het dè plaats om degene te bedanken die op een directe of indirecte wijze betrokken zijn geweest bij de totstandkoming van dit rapport. Een aantal van hen wil ik hier met name noemen.

In de eerste plaats Jan Kors, onder wiens deskundige begeleiding ik met veel plezier aan het afstudeeronderzoek heb kunnen werken. Jan, bedankt voor het vertrouwen dat je in mij stelde. De vrijheid die je mij hebt gegeven om mijn onderzoek uit te voeren en jouw directe betrokkenheid, heb ik als bijzonder prettig en waardevol ervaren. Jouw wijze van formuleren was zeer leerzaam, waarvoor mijn dank. Daarnaast was Arie Hasman als afstudeerhoogleraar op de achtergrond aanwezig met zijn advies en kritische wijze van begeleiden. In het bijzonder wil ik U bedanken voor de snelle beoordeling van dit rapport. Ook waardeer ik de enthousiaste bijdrage die Bob Schijvenaars heeft geleverd, speciaal voor wat betreft de technische assistentie.

Naast de mensen die direct bij het afstudeeronderzoek betrokken waren, wil ik ook mijn familie en vrienden bedanken voor hun belangstelling en aanmoedigingen. Erg belangrijk voor mij was de ondersteuning en het vertrouwen in een goede afloop die mijn ouders mij gaven. Vooral gedurende de periode na het plotselinge overlijden van mijn moeder, waarin mijn vader langdurig in het ziekenhuis verbleef, heb ik aan de mensen om mij heen veel te danken gehad. Ook vanaf het moment dat bij mij de darmziekte *colitis ulcerosa* werd vastgesteld hebben diverse mensen mij met zorg omgeven. Zonder iemand te kort te willen doen, wil ik een aantal van hen hier dankzeggen.

Allereerst bedank ik mijn vader voor zijn interesse en aanmoedigingen om dit rapport in deze voor mij moeilijke tijd te voltooien. Pa, ondanks de aangrijpende omstandigheden waarin U zich bevond, zijn Uw morele en financiële ondersteuning van onschatbare waarde geweest. Daarnaast waren mijn broer Hans en zijn vrouw Joke altijd aanwezig om mij daar waar nodig ter zijde te staan.

Voorts bedank ik mijn vrienden Elmar Franken en Marcel Meesters voor de interesse die zij toonden en de tijd die zij spendeerden aan het bestuderen van dit rapport.

Mijn welgemeende waardering gaat uit naar mijn collega's bij Van Exter Polak & Charlouis, alwaar ik sinds begin dit jaar als octrooigemachtigde i.o. werkzaam ben. In het bijzonder noem ik mijn mentor Hans Mertens, voor wiens vertrouwen en begrip ik zeer erkentelijk ben. De familiaire betrokkenheid die ik ondervind heeft op mij grote indruk gemaakt en mij doen inzien dat het leven niet alleen uit (afstudeer)werk bestaat. Ik verheug mij er op aan de nieuwe uitdagingen op creatieve wijze invulling te geven.

Lieve Karen, als ik iemand te kort doe met woorden dan ben jij het wel. Je bent voor mij een onmisbare en niet aflatende steun geweest. De liefde en het geduld waarmee jij mij hebt omringd, hebben mij door de afgelopen periode geloodst en gestimuleerd om dit rapport af te ronden. Zonder jou was het nog moeilijker geweest. Mijn waardering hiervoor heb ik vast niet altijd even duidelijk laten blijken en ik hoop dat ik je daarmee niet te kort heb gedaan. Bedankt dat jij er altijd voor mij bent.

ASWIN HOFFMANN

Rotterdam, september 1996

Inhoudsopgave

Voorwoord v

Samenvatting ix

Abstract xi

1 Inleiding 1

2 Inductieve leertechnieken 5
- 2.1 Inleiding . . . 5
- 2.2 Schatting van prestatiematen . . . 5
  - 2.2.1 Diagnostische prestatiematen . . . 6
  - 2.2.2 Methoden voor schatting van de werkelijke prestaties . . . 7
- 2.3 Inductie van beslisbomen . . . 10
  - 2.3.1 Constructie van een beslisboom . . . 12
- 2.4 Inductie van beslisregels . . . 17
  - 2.4.1 Generatie van produktieregels uit beslisbomen . . . 18
  - 2.4.2 Inductie van de enkele 'beste' regel . . . 19
- 2.5 Conclusies . . . 24

3 Ontwikkeling van een nieuw algoritme 25
- 3.1 Inleiding . . . 25
- 3.2 Uitputtend zoeken in beslisbomen . . . 26
  - 3.2.1 Een methode voor het genereren van beslisbomen . . . 27

3.2.2 Complexiteitsanalyse . . . 27
3.3 Uitputtend zoeken in beslisregels . . . 32
3.3.1 Een methode voor het genereren van beslisregels . . . 32
3.3.2 Complexiteitsanalyse . . . 32
3.3.3 De doelmatigheid van beslisregels . . . 34
3.3.4 Heuristieken . . . 35
3.3.5 Het bepalen van de optimale regellengte . . . 40
3.4 Conclusies . . . 41

4 Experimenten **43**
4.1 Inleiding . . . 43
4.2 Selectie van gegevensbestanden . . . 43
4.3 Optimaliteit van geïnduceerde beslisregels . . . 45
4.3.1 Uitputtend zoeken naar optimale regels . . . 45
4.3.2 Uitputtend versus heuristisch zoeken naar optimale regels . . . 48
4.4 Vergelijking met conventionele inductietechnieken . . . 52
4.4.1 Maximalisatie van de classificatienauwkeurigheid . . . 52
4.4.2 Optimalisatie van de diagnostische prestaties . . . 55
4.5 Conclusies . . . 58

5 Conclusies **61**

6 Aanbevelingen **63**

Bibliografie **65**

A Complexiteitsanalyse **69**
A.1 Topologieën van binaire bomen . . . 69
A.2 Gelabelde topologieën van binaire bomen . . . 72
A.3 Logisch unieke *k*-conjuncties . . . 75

B Geselecteerde gegevensbestanden **79**
B.1 Inleiding . . . 79
B.2 Specificaties . . . 79
B.2.1 BU: Bupa leverstoornissen . . . 79
B.2.2 GL2: Glass identificatie . . . 80
B.2.3 HE: Hepatitis domein . . . 81
B.2.4 IR: Iris domein . . . 82
B.2.5 PI: Pima Indians Diabetes domein . . . 83
B.3 Overzicht van eerder gepubliceerde resultaten . . . 84

C Programmatuur **85**
C.1 Inleiding . . . 85
C.2 Beschrijving van bestanden . . . 85
C.3 Inventaris van programmacode . . . 91

Samenvatting

Uit de patroonherkenning en de kunstmatige intelligentie zijn empirische leertechnieken bekend om op automatische wijze, aan de hand van een verzameling voorbeeldobjecten, een classificator te construeren. Voor toepassing binnen de medische diagnostiek is het van groot belang dat de resulterende classificator "begrijpelijk" is, dat wil zeggen dat artsen de werking ervan intuïtief begrijpen en deze aansluit bij de wijze waarop zij gewend zijn te redeneren. Beslisbomen en beslisregels voldoen aan deze eis, waardoor het brede veld aan technieken beperkt wordt tot machineleertechnieken voor het inductief leren van beslisboom- of beslisregelclassificatoren. Voor toepassing binnen de geneeskunde zijn vrijwel alle inductietechnieken behept met (tenminste) twee nadelen. Het eerste nadeel is dat vrijwel altijd getracht wordt de classificatienauwkeurigheid te maximaliseren, terwijl diagnostische prestatiematen als sensitiviteit en specificiteit veelal relevanter zijn. Het tweede nadeel is dat de optimaliteit van de resulterende classificator niet gegarandeerd is.

Doel van dit onderzoek is, zo mogelijk op grond van reeds bestaande inductietechnieken, een methode te ontwikkelen waarmee het mogelijk is een classificator te construeren die optimaal is onder voorwaarden die aan de diagnostische prestatiematen van de resulterende classificator zijn opgelegd.

Allereerst is een uitgebreid literatuuronderzoek verricht naar de mogelijkheden die conventionele inductietechnieken bieden. Daaruit blijkt dat de diagnostische prestatiematen van de resulterende classificator tijdens het inductieproces moeilijk of niet te sturen zijn, doordat de relatie tussen heuristische evaluatiefuncties die bij het inductieproces

worden gebruikt en de te optimaliseren prestatiemaat onduidelijk is. Daarenboven kan globale optimaliteit van de resulterende classificator niet worden gegarandeerd, doordat het stapsgewijs verlopende inductieproces op lokaal optimale beslissingen is gebaseerd.

Meer uitgebreide zoektechnieken bieden een oplossing om beide nadelen te ondervangen. Uit een analyse van het uitputtend genereren van beslisboom- en beslisregelclassificatoren, blijkt dat het genereren van beslisregels efficiënter is. Er is een nieuwe techniek ontwikkeld waarmee het mogelijk is de enkele beste regel te induceren die één van de diagnostische prestatiematen maximaliseert onder oplegging van restricties aan de andere maten. Door de enorme rekenkundige complexiteit, blijkt het in de praktijk vaak nodig heuristieken te gebruiken. Er is een aantal eenvoudige heuristieken voorgesteld en geïmplementeerd. Eveneens is aangegeven hoe deze heuristieken op de complexiteit ingrijpen.

Het nieuwe algoritme, EXPLORE (Exhaustive Procedure for Logic-Rule Extraction) genaamd, is toegepast op een aantal eerder in de literatuur geanalyseerde gegevensbestanden. Uit experimentele resultaten blijkt dat een aantal van de voorgestelde heuristieken de complexiteit sterk reduceert, terwijl het algoritme er nog steeds in slaagt een optimale of sub-optimale oplossing te vinden. Vergeleken met andere, meer conventionele inductietechnieken blijkt dat wanneer als optimaliseringscriterium maximalisatie van de classificatienauwkeurigheid wordt gebruikt, er vergelijkbare of betere classificatoren worden geconstrueerd.

Voorts blijkt uit een vergelijking van EXPLORE met het alom bekende CART-algoritme voor inductie van beslisbomen, dat er een duidelijke uitwisseling tussen snelheid en optimaliteit bestaat. De hiërarchische benadering, die kenmerkend is voor beslisbomen, resulteert in een sneller leerproces, terwijl volgens de nieuwe methode het uitvoeriger combineren van kenmerktests de optimaliteit van de oplossing ten goede komt.

In de huidige implementatie van EXPLORE is het mogelijk al naar gelang de complexiteit van de classificatietaak uitputtend te zoeken of één of meer heuristieken te gebruiken. De experimentele resultaten verschaffen aanwijzingen dat meer gecompliceerde heuristieken die gebaseerd zijn op een *beam search* strategie of een meerstapsanalyse, het inductieproces aanzienlijk kunnen versnellen. Daarnaast zou nader onderzoek zich op strategieën voor een gevoeligheidsanalyse en meerklassenproblemen kunnen richten.

Alhoewel de praktische bruikbaarheid van EXPLORE momenteel beperkt is tot het induceren van korte regels, is zij veelal in het voordeel wanneer een goede oplossing in de vorm van een korte regel bestaat.

In dit rapport is aangetoond dat, in tegenstelling tot de meeste andere inductietechnieken, het met de nieuwe methode mogelijk is de diagnostische prestaties van de resulterende classificator aan te passen aan de specifieke voorwaarden die de gebruiker oplegt. Zodoende is een methode gecreëerd om toepassings-specifieke beslisregels te construeren die optimaal zijn.

Abstract

Pattern recognition and artificial intelligence provide empirical learning techniques for automated construction of a classifier based on a set of sample objects. When such a classifier is to be applied for medical diagnosis it is essential that it is 'comprehensible', i.e. physicians should intuitively understand its functioning and it should mimic the physicians' reasoning. Decision trees and decision rules comply with this requirement, which reduces the wide range of techniques to those of machine learning techniques for inductive learning of decision-tree or decision-rule classifiers.

Nearly all conventional induction techniques have (at least) two important drawbacks as methods for constructing classifiers in medical science. First, techniques almost invariably try to maximize classification accuracy, whereas in a medical context diagnostic performance measures such as sensitivity and specificity are often more important. Second, it is impossible to ensure the optimality of the resulting classifier.

The aim of the present study was to develop, if possible on basis of existing induction techniques, a method that automatically constructs a classifier which is optimal while fulfilling user-imposed conditions on the diagnostic performance measures of the resulting classifier.

First, a comprehensive inquiry has been made into literature on the possibilities offered by conventional induction techniques. The results show it is difficult or even impossible to control the diagnostic performance measures of the resulting classifier during the induction process, due to indirect relationship between heuristic evaluation functions used for induction and the performance measure to be optimized. Moreover,

it is impossible to guarantee global optimality of the resulting classifier, because the step-by-step induction process is based on locally optimal decisions.

More extensive searching techniques provide a solution for elimination of these drawbacks. An analysis was therefore made of exhaustive generation of decision-tree and decision-rule classifiers. It proved that the generation of decision rules is more efficient. A new technique for induction of the single best decision rule which maximizes one of the diagnostic performance measures subject to constraints on the other measures was implemented and applied. Due to the computational complexity of an exhaustive search, in practice there appears to be a need for heuristics reducing the search space to manageable proportions. Several uncomplicated heuristics were proposed and implemented. Implications thereof have been discussed in relation to complexity.

The new algorithm, called EXPLORE (Exhaustive Procedure for Logic-Rule Extraction), was used on data sets previously analyzed in literature. Experimental results show that various of the heuristics suggested can greatly reduce complexity while the algorithm is still able to find an optimal or near-optimal solution. A comparison of EXPLORE with alternative induction techniques proves that maximizing classification accuracy allows the induction of competitive decision rules.

Furthermore, a comparison of EXPLORE with the widely known CART algorithm for decision-tree induction reveals there is a manifest trade-off between learning speed and optimality. The hierarchical approach, which is characteristic of decision trees, yields a higher learning speed, whereas one of the merits of the new method is that the more exhaustive combination of feature tests is conducive to the optimality of the solution.

In its current implementation, EXPLORE can use uncomplicated heuristics, depending on the complexity of the classification task. There is empirical evidence that more sophisticated heuristics based on a beam search strategy and multi-stage analysis can greatly speed up the induction process. This is a field of further research. Other areas of future research may focus on sensitivity analysis and multi-class problems.

Although the practical usefulness of EXPLORE is currently limited to the induction of short-length decision rules, it mostly has a competitive edge when a suitable solution in the form of a short-length rule is available.

This thesis shows that the new method, unlike most other conventional induction techniques, enables the diagnostic performance measures of the resulting classifier to be tuned to specific constraints imposed by the user. Thus, an opportunity has been created to construct application-specific decision rules that are optimal.

HOOFDSTUK 1

Inleiding

Aanleiding voor het onderzoek

Het nemen van beslissingen is een probleem dat binnen de algemene categorie van classificatie valt [Clancey, 1985]. Classificatie binnen de geneeskunde heeft betrekking op het klassieke medisch diagnoseprobleem: op grond van een verzameling observaties wordt de patiënt bij één van een aantal mogelijke ziektecategorieën ingedeeld. Alhoewel de expert door jarenlange training en ervaring beschikt over kennis om dergelijke beslissingen te nemen en te onderbouwen, kunnen bij het uitvoeren van deze patroonherkenningstaak fouten worden gemaakt.

Sinds de vijftiger jaren wordt daarom onderzoek verricht naar mogelijkheden van computers om experts te ondersteunen bij het nemen van beslissingen. Heuristische expertkennis dient daartoe in een formeel classificatiemodel te worden vervat. Daarbij doet zich het probleem voor dat expertise schaars is, acquisitie en formalisatie van kennis een moeizaam proces is, het formalisatieproces niet garandeert dat de resulterende classificator optimaal is met betrekking tot een of andere prestatiemaat en het onduidelijk is op welke wijze de classificator geoptimaliseerd kan worden.

Met de opkomst van computers bleek het mogelijk een deel van de problemen te ondervangen door technieken te ontwikkelen die het mogelijk maken om zonder inmenging van een expert classificatoren te construeren. Dergelijke leertechnieken werken met een model waarbij het 'leren' bestaat uit het aanpassen van parameters binnen de modelstructuur op grond van een verzameling voorbeeldobjecten, waarbij elk object bestaat uit een patroon van kenmerken en een corresponderend klasselabel. Het leerproces kan

worden beschouwd als een generalisatieproces over de empirische associaties tussen kenmerkpatroon en klasselabel binnen de beperkingen die door het classificatiemodel worden opgelegd. Een dergelijk leerproces wordt aangeduid met de term *supervised learning*—leren met een leermeester—waarbij het de taak van de leermeester is de juiste klasselabels toe te kennen.

Leertechnieken voor classificatie worden onderverdeeld in twee categorieën [James, 1985]. Een eerste categorie bestaat uit parametrische en niet-parametrische technieken afkomstig uit de statistische patroonherkenning [Fukunaga, 1972; Duda & Hart, 1973], waaronder ook kunstmatige neurale netwerken. Een tweede bestaat uit machineleertechnieken voor het inductief leren van beslisbomen [Breiman *et al.*, 1984; Quinlan, 1986] of beslisregels [Michalski *et al.*, 1986; Quinlan, 1987a]. Technieken uit beide categorieën zijn doorgaans toepasbaar op hetzelfde probleem, maar zijn verschillend in hun onderliggend model en de manier waarop het classificatieformalisme wordt gepresenteerd. Een voordeel van beslisbomen en -regels is dat ze een modulaire opzet hebben, intuitief begrijpelijk zijn en nauw aansluiten bij de manier waarop experts gewend zijn te redeneren [Weiss & Kapouleas, 1989]. Voor toepassing binnen de geneeskunde is het van groot belang dat de classificator 'begrijpelijk' is. De nadruk ligt daarom op inductieve leertechnieken.

Inductie-algoritmen worden met succes toegepast voor tal van classificatietaken [Swain & Hauska, 1977; Quinlan, 1990; Kern *et al.*, 1993]. Voor toepassing binnen de geneeskunde zijn deze technieken echter behept met een tweetal nadelen. Een eerste nadeel is dat vrijwel altijd getracht wordt de foutfractie van de classificator te minimaliseren, omdat dit binnen de patroonherkenning een veelgebruikte prestatiemaat is. Voor medisch-diagnostische beslisproblemen zijn echter vaak andere prestatiematen relevant [Galen & Gambino, 1975]. Met conventionele inductietechnieken zijn deze diagnostische prestatiematen moeilijk te controleren. Daarnaast is het wenselijk dat de gebruiker voorwaarden kan stellen aan de diagnostische prestaties van de resulterende classificator. Voor hartfunctiediagnostiek kan het bijvoorbeeld van belang zijn een classificator te hebben die geen enkel infarct mist, terwijl in zo min mogelijk gevallen ten onrechte tot infarct wordt beslist. Een tweede nadeel is dat in het stapsgewijs verlopende inductieproces lokaal optimale beslissingen worden genomen, waarmee globale optimaliteit van de resulterende classificator niet gegarandeerd is.

Een inductietechniek waarmee het mogelijk is een classificator te construeren die optimaal is onder voorwaarden die aan de diagnostische prestaties van de resulterende classificator zijn opgelegd, was eerder niet bekend. De beschikking over een dergelijke techniek zou het mogelijk maken tal van medisch-diagnostische beslisprocedures te optimaliseren.

Doelstelling van het onderzoek

Het doel van dit onderzoek is, zo mogelijk op grond van reeds bestaande inductietechnieken, een methode te ontwikkelen waarmee de gebruiker de prestaties van de resulterende classificator op een eenduidige wijze kan beïnvloeden.

In eerste instantie worden daartoe aan de hand van een literatuuronderzoek de mogelijkheden van conventionele inductie-algoritmen onderzocht. Op grond hiervan worden overwegingen gegeven om al dan niet een nieuwe methode te ontwikkelen. Wanneer dit nodig blijkt te zijn, dient in tweede instantie een mogelijke oplossing daarvoor te worden aangedragen. De methode dient vervolgens te worden geëvalueerd aan de hand van uit de literatuur bekende gegevensbestanden.

Structuur van het rapport

De opbouw van dit rapport is als volgt. In hoofdstuk 2 wordt een globale beschrijving gegeven van machineleertechnieken voor het inductief leren van beslisbomen en beslisregels. Tevens zal worden ingegaan op methoden voor het schatten van prestatiematen. Op basis hiervan wordt duidelijk gemaakt wat de tekortkomingen van de conventionele inductietechnieken zijn en worden de overwegingen gegeven om een geheel nieuwe inductiemethode te ontwikkelen. In hoofdstuk 3 komt de nieuwe methode aan de orde. Een gedetailleerde beschrijving van het onderliggende model wordt gepresenteerd, alsmede een analyse van de rekenkundige complexiteit van het algoritme. In aansluiting hierop worden heuristieken geïntroduceerd, met als doel de complexiteit te reduceren. Aangegeven wordt hoe de diverse heuristieken ingrijpen op de afzonderlijke termen van de complexiteitsformule. In hoofdstuk 4 worden experimenten met uit de literatuur bekende gegevensbestanden beschreven en worden de resultaten, waar mogelijk, vergeleken met die van andere, meer conventionele inductietechnieken. Hoofdstuk 5 bevat de conclusies van het onderzoek en opmerkingen met betrekking tot de bruikbaarheid van de ontwikkelde methode voor praktische toepassingen. Tenslotte worden in hoofdstuk 6 aanbevelingen gedaan voor verder onderzoek.

HOOFDSTUK 2

Inductieve leertechnieken

2.1 INLEIDING

In de literatuur zijn vele technieken voor het inductief leren van beslisboom- en beslisregelclassificatoren uitvoerig beschreven. In dit hoofdstuk volgt een algemene uiteenzetting van de fundamentele principes die ten grondslag liggen aan de diverse technieken. Aangegeven zal worden hoe beslisbomen en beslisregels op grond van een verzameling voorbeeldobjecten kunnen worden geconstrueerd, met als doel na te gaan of deze conventionele technieken geschikt zijn om een classificator te construeren die optimaal is onder voorwaarden die aan de diagnostische prestatiematen zijn opgelegd.

Alvorens daarop in te gaan, komt in de volgende paragraaf eerst het evalueren van de prestaties van een classificator aan de orde. Bovendien worden hier een aantal prestatiematen geïntroduceerd die binnen de medische diagnostiek relevant zijn.

2.2 SCHATTING VAN PRESTATIEMATEN

Het doel van een lerend systeem is om op grond van een verzameling voorbeeldobjecten een classificator te construeren die succesvol is in het classificeren van nieuwe objecten waarvan de klasse *a priori* onbekend is. De mate van succes waarmee de classificator zijn taak uitvoert kan tot uitdrukking worden gebracht in één of meer prestatiematen.

Een binnen de patroonherkenning veelgebruikte prestatiemaat is de foutfractie, die het aantal misclassificaties deelt door het totaal aantal geclassificeerde objecten [Fuku-

naga, 1972; Duda & Hart, 1973]. De geneeskunde kent daarnaast een aantal andere maten om de prestaties van een diagnostische test te beschrijven [Galen & Gambino, 1975; Connell & Koepsell, 1985]. Deze kunnen eveneens worden toegepast bij het evalueren van classificatoren. Vijf van dergelijke diagnostische prestatiematen worden in de volgende subparagraaf meer in detail beschreven.

2.2.1 Diagnostische prestatiematen

Binnen de geneeskunde komt het classificatieprobleem vaak neer op een dichotomie, waarbij onderscheid gemaakt wordt tussen de klassen 'ziek' en 'gezond'. De classificator is dan een diagnostische test die een positief of negatief resultaat geeft. Er doen zich hierbij vier mogelijke beslisuitkomsten voor: correct-positief, fout-positief, fout-negatief en correct-negatief. Deze resultaten kunnen in een classificatiematrix worden weergegeven. Een dergelijke matrix is als tabel 2.1 afgebeeld. De getallen *CP*, *FP*, *FN* en *CN* staan respectievelijk voor het aantal correct-positieve, fout-positieve, fout-negatieve en correct-negatieve beslissingen.

Tabel 2.1 Classificatiematrix voor een dichotome test.

Test-	Ziekte		
resultaat	Aanwezig	Afwezig	Totaal
Positief	CP	FP	$CP + FP$
Negatief	FN	CN	$FN + CN$
Totaal	$CP + FN$	$FP + CN$	$CP + FP + FN + CN$

De in de medische diagnostiek meest bekende maten om de prestaties van een dichotome test tot uitdrukking te brengen zijn: sensitiviteit, specificiteit, positief predictieve waarde, negatief predictieve waarde en nauwkeurigheid [Galen & Gambino, 1975]. Sensitiviteit en specificiteit geven de 'gevoeligheid' van de test voor een klasse weer; in medische termen is sensitiviteit de kans op een positief testresultaat bij aanwezigheid van de ziekte en specificiteit de kans op een negatief testresultaat bij afwezigheid van de ziekte. De predictieve waarde geeft de 'voorspelbaarheid' van een klasse voor een testresultaat weer; in medische termen is de positief predictieve waarde de kans op aanwezigheid van de ziekte bij een positief testresultaat en de negatief predictieve waarde de kans op afwezigheid van de ziekte bij een negatief testresultaat. Nauwkeurigheid geeft de 'correctheid' van de test aan; het is de kans op een correcte beslissing.

Uit tabel 2.1 kunnen schattingen van deze prestatiematen worden afgeleid. De definities daarvan zijn in tabel 2.2 weergegeven. De genoemde diagnostische prestatiematen

Tabel 2.2 Schatters van diagnostische prestatiematen.

Sensitiviteit	$CP / (CP + FN)$
Specificiteit	$CN / (FP + CN)$
Positief predictieve waarde	$CP / (CP + FP)$
Negatief predictieve waarde	$CN / (FN + CN)$
Nauwkeurigheid	$(CP + CN) / (CP + FP + FN + CN)$

zijn alle correctheidsfracties. De corresponderende complementaire foutfracties worden doorgaans buiten beschouwing gelaten, met uitzondering van de hiervoor gedefinieerde foutfractie, die gelijk is aan 1 – nauwkeurigheid.

Opgemerkt dient te worden dat de hiervoor beschreven prestatiematen een meer algemene geldigheid hebben dan voor de geneeskunde alleen. De methoden voor schatting van prestatiematen die in de volgende paragraaf worden beschreven, zijn algemeen van aard en kunnen elk voor bepaling van de genoemde prestatiematen worden aangewend.

2.2.2 Methoden voor schatting van de werkelijke prestaties

De *werkelijke* prestatie[1] van een classificator is gedefinieerd als de prestatie van de classificator wanneer deze zou worden geëvalueerd op de totale populatie. In de praktijk is de totale populatie echter vrijwel altijd onbekend, waardoor slechts *empirische* prestaties bepaald kunnen worden op grond van een beperkte verzameling (steekproef)objecten. De vraag die zich hierbij voordoet is in hoeverre empirische prestaties een 'goede' schatting van de werkelijke prestaties opleveren. Een schatting dient immers precies (d.w.z. *unbiased*) te zijn en een hoge mate van nauwkeurigheid (d.w.z. een lage variantie) te hebben.

Niet-parametrische technieken voor schatting van de werkelijke prestaties zijn uitvoerig bestudeerd vanuit de statistiek [Toussaint, 1974; Efron & Gong, 1983] en de patroonherkenning [Fukunaga, 1972; Duda & Hart, 1973]. Daarbij dient te worden opgemerkt dat al deze technieken een zekere vertekening introduceren: sommige zijn optimistisch (d.w.z. neigen de waarde van de prestatiemaat te overschatten), terwijl andere pessimistisch zijn (d.w.z. neigen de waarde van de prestatiemaat te onderschatten).

De essentiële voorwaarde voor een goede schatting van de werkelijke prestaties van een classificator is, dat de verzameling objecten waarmee de prestaties van de classificator worden geëvalueerd, een aselecte (*random*) steekproef uit de populatie moet zijn. Dit betekent dat er geen objecten tot de verzameling mogen behoren die een zekere 'voorkeur' genieten boven andere objecten en dat een expert zich niet bezig mag houden met het selecteren van 'representatieve' objecten.

[1] In het vervolg wordt met "prestatie" de waarde van de prestatiemaat aangeduid.

In de volgende paragrafen worden vier methoden beschreven om een schatting van de werkelijke prestaties van een classificator te maken. De nomenclatuur is overgenomen uit [Toussaint, 1974].

R methode: schatting door resubstitutie

Een voor de hand liggende methode om een schatting van de prestaties van een classificator te maken is de prestatiemaat te bepalen met de verzameling voorbeeldobjecten die gebruikt is om de classificator te construeren. Deze zogenaamde resubstitutieschatting van de werkelijke prestatiemaat is echter geneigd een te optimistische vertekening te geven [Duda & Hart, 1973]. Door 'overspecialisatie' van de classificator op de specifieke eigenschappen van de voorbeeldobjecten kunnen de werkelijke prestaties op nieuwe objecten slechter uitvallen dan de prestaties op de voorbeeldobjecten suggereren. Om een betere schatting van de prestaties te krijgen, is het dus gewenst de classificator te evalueren op een verzameling gelabelde objecten die niet gebruikt is voor de constructie van de classificator.

H methode: schatting door partitie in leer- en testverzameling

Bij de zogenaamde *holdout* methode wordt de verzameling van gelabelde objecten in een leer- en testverzameling gepartitioneerd. De leerverzameling wordt uitsluitend gebruikt om de classificator te construeren. De testverzameling wordt gebruikt om de prestaties van de classificator te bepalen.

In de praktijk wordt veelal tweederde van de objecten gebruikt voor constructie van de classificator en éénderde voor testen. Voor grote aantallen (> 1000) testobjecten is dit een aanvaardbare methode [Highleyman, 1962]. Bij objectverzamelingen van geringere omvang doet zich het probleem voor dat er onvoldoende testobjecten beschikbaar zijn om een betrouwbare schatting te kunnen maken. Highleyman (1962) toont aan dat de schatting niet alleen een te pessimistische vertekening geeft, maar ook een grote variantie heeft.

Daarnaast maakt deze methode geen efficiënt gebruik van de beschikbare objecten. Slechts een beperkt gedeelte van de objecten wordt daadwerkelijk gebruikt om de classificator te construeren, terwijl de objecten uit de testverzameling eveneens nuttige informatie kunnen bevatten. Dit is vooral voor kleinere objectverzamelingen nadelig.

Random subsampling

Om de betrouwbaarheid en efficiëntie te vergroten kan de geschatte prestatiemaat worden gemiddeld over mogelijke partities van vaste grootte [Duda & Hart, 1973]. Daartoe wordt de objectverzameling meerdere malen steekproefsgewijs in een leer- en testverzameling gepartitioneerd, waarbij telkens een classificator wordt geconstrueerd. De prestatiemaat wordt geschat door middeling over de afzonderlijke prestatiematen die zijn afgeleid

uit de onafhankelijk en willekeurig gegenereerde partities. Alhoewel de objecten voor deze schatting efficiënter worden gebruikt dan bij de H methode, wordt telkens nog altijd een beperkt gedeelte gebruikt om een classificator te construeren. Bovendien is het uiteindelijke resultaat nog steeds pessimistisch vertekend.

*Resampling*methoden gebruiken de objecten op nog efficiëntere wijze en geven een kleinere vertekening van de geschatte prestaties ten opzichte van de voorgaande methoden. In het navolgende worden twee resamplingmethoden beschreven, waarbij nader wordt ingegaan op de wijze waarop de objectverzameling wordt gepartitioneerd.

U methode: schatting door *leave-one-out*

Een elegante en ongecompliceerde techniek om de werkelijke prestaties van een classificator te schatten is de *leave-one-out* methode [Lachenbruch & Mickey, 1968].

Wanneer de objectverzameling bestaat uit n objecten, dan wordt de classificator geconstrueerd op grond van $n-1$ objecten en getest op het resterende object. Deze procedure wordt n maal uitgevoerd door achtereenvolgens elk van de n objecten uit te sluiten van de leerverzameling. Op deze wijze wordt op elk object één maal getest en worden telkens op één na alle objecten gebruikt om de classificator te construeren. De geschatte prestatiemaat wordt verkregen door middeling over de n afzonderlijke prestatiematen.

Uit de literatuur is bekend dat deze benadering zeer acceptabele resultaten geeft [Lachenbruch & Mickey, 1968; Fukunaga, 1972]. De *leave-one-out* schatter geeft een nagenoeg onvertekend beeld van de werkelijke prestatiemaat; zelfs voor geringe omvang (≈ 100) van de objectverzameling is de schatting precies. Desondanks kent deze methode tenminste twee nadelen.

Een eerste nadeel betreft de nauwkeurigheid. De variantie van de *leave-one-out* schatter is groot, vooral wanneer de omvang van de objectverzameling gering is. Een methode die minder precies, maar veel nauwkeuriger is, kan in dit geval de voorkeur hebben. Een tweede, praktisch nadeel is dat deze methode zeer rekenintensief is voor omvangrijke objectverzamelingen, omdat n maal een classificator geconstrueerd moet worden. Dit probleem kan worden ondervangen door de omvang van leer- en testverzameling respectievelijk te laten af- en toenemen.

Π methode: schatting door *cross-validation*

De *leave-one-out* methode is een speciaal geval van de algemene klasse van *cross-validation* technieken [Stone, 1974]. Bij k-voudige *cross-validation* worden de objecten aselect verdeeld over k disjuncte partities van gelijke omvang. Elk van de k partities fungeert achtereenvolgens als testverzameling, terwijl de resterende objecten steeds als leerverzameling dienen. De uiteindelijke prestatiemaat wordt verkregen door middeling over de k onafhankelijke prestatiematen die voor elke partitie zijn bepaald.

Empirische resultaten tonen aan dat 10-voudige *cross-validation* adequaat en nauwkeurig is, speciaal voor omvangrijke (> 100) objectverzamelingen waarvoor de *leave-*

one-out methode te rekenintensief is [Breiman *et al.*, 1984]. De methode kan worden verfijnd door de objecten in de leer- en testverzamelingen te stratificeren, zodanig dat het percentage van elke klasse die van de gehele objectverzameling benadert.

Opgemerkt dient te worden dat voor $k = n$ de Π methode in essentie reduceert tot de U methode. Doorgaans is k echter vele malen kleiner dan n, waardoor de Π methode aanzienlijk minder iteraties vergt dan de U methode en bovendien een kleinere vertekening introduceert dan de H methode. De Π methode is derhalve een compromis tussen de H en U methode en is zodoende bij uitstek geschikt voor objectverzamelingen van 'gemiddelde' omvang.

Resamplingtechnieken verschaffen betrouwbare schattingen van de werkelijke prestaties van een classificator. Nagenoeg alle objecten worden gebruikt voor constructie van de classificator en alle objecten worden gebruikt voor testen. Uitgebreide analyses en simulaties met verschillende classificatietechnieken laten zien dat deze schattingen nauwelijks vertekenen ten opzichte van schattingen die gebaseerd zijn op nieuwe objecten. Een ander voordeel van resampling is dat het onderzoekers in staat stelt om experimenten met dezelfde objecten te herhalen en de resultaten daarvan met eerder gepubliceerde resultaten te vergelijken, zonder dat afwijkingen ten opzichte hiervan hoofdzakelijk aan de natuurlijke variabiliteit van de partitie in een enkel (*random*) leer- en testexperiment kunnen worden toegeschreven.

Tabel 2.3 geeft nog eens beknopt weer hoe de verschillende technieken een verzameling van n objecten gebruiken om de prestaties te schatten.

Tabel 2.3 Methoden voor schatting van prestatiematen.

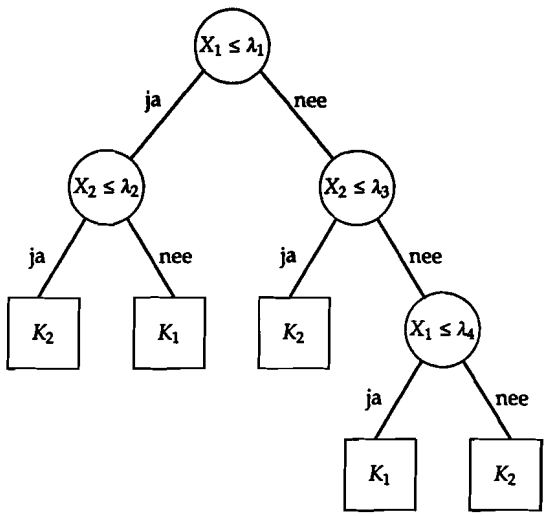
	Resubstitutie	Holdout	Leave-one-out	k-fold cross-validation
leerobjecten	n	j	$n-1$	$\frac{k-1}{k}n$
testobjecten	n	$n-j$	1	$\frac{n}{k}$
iteraties	1	1	n	$1 \leq k \ll n$

2.3 INDUCTIE VAN BESLISBOMEN

Beslisbomen bieden een effectieve implementatie van hiërarchische classificatoren. Dankzij hun conceptuele eenvoud en hun rekenkundige effectiviteit, vormen zij verreweg de meest ontwikkelde techniek om objecten te classificeren met behulp van een verzameling beslisregels die de hele beslisruimte dekt.

Door de jaren heen is een grote verscheidenheid aan methoden ontstaan om beslisbomen te ontwerpen [Henrichson & Fu, 1969; Meisel & Michalopoulos, 1973; Friedman, 1977; Payne & Meisel, 1977; Sethi & Sarvarayudu, 1982; Argentiero *et al.*, 1982; Breiman

et al., 1984]. Deze methoden worden met succes toegepast op problemen als *remote sensing* [Swain & Hauska, 1977], karakterherkenning [Wang & Suen, 1984] en medische diagnose en prognose [Mui & Fu, 1980; Landeweerd *et al.*, 1983; Lin & Fu, 1983].



Figuur 2.1 Voorbeeld van een binaire beslisboom.

Een beslisboom bestaat uit een verzameling knooppunten die in een hiërarchische structuur door takken met elkaar zijn verbonden. Een karakteristieke binaire beslisboom is weergegeven in figuur 2.1. De als cirkels afgebeelde knopen zijn beslisknopen, waarvan de descendenten worden bepaald door een drempelwaarde λ_i op een specifiek kenmerk X_i. Opgemerkt dient te worden dat een kenmerk in meerdere beslisknopen kan voorkomen. De als vierkanten afgebeelde knopen zijn eindknopen, waaraan een klasselabel K_i wordt toegewezen. Op grond van de uitslag waarbij een kenmerkwaarde tegen een drempelwaarde wordt getest, wordt de linker- of rechtertak gekozen. Indien een onbekend object voor classificatie wordt aangeboden, dan krijgt het object het klasselabel toegewezen van de eindknoop waarin het uiteindelijk belandt.

Voor elke boom geldt dat het pad dat leidt tot een eindknoop correspondeert met een beslisregel die bestaat uit de conjunctie van de betreffende tests op het pad in de boom. Zo correspondeert het pad naar de meest linkse eindknoop van de boom in figuur 2.1 met de regel (2.1).

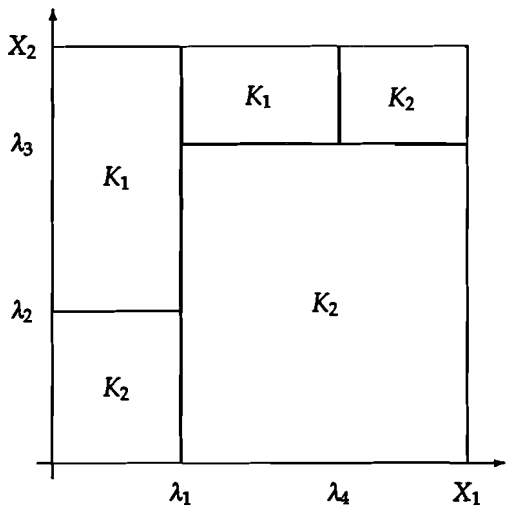
$$\textbf{als } (X_1 \leq \lambda_1) \text{ EN } (X_2 \leq \lambda_2) \textbf{ dan } K_2 \tag{2.1}$$

Wanneer er voor een bepaalde klasse meerdere paden in de boom bestaan, corresponderen zij met disjuncties in de beslisregel. Regel (2.2) correspondeert met de twee paden die leiden tot een eindknoop met klasselabel K_1.

$$\textbf{als } (X_1 \leq \lambda_1) \text{ EN } (X_2 > \lambda_2) \quad \text{OF} \quad (X_1 > \lambda_1) \text{ EN } (X_2 > \lambda_3) \text{ EN } (X_1 \leq \lambda_4) \textbf{ dan } K_1 \tag{2.2}$$

2.3.1 Constructie van een beslisboom

Om een beslisboom te construeren wordt aangenomen dat een verzameling van gelabelde objecten beschikbaar is. De beslisboom wordt dan geconstrueerd door de kenmerkruimte recursief te partitioneren en de resulterende subgebieden met de overeenkomstige klassen te laten corresponderen. In figuur 2.2 is een partitie van de tweedimensionale kenmerkruimte weergegeven volgens de in figuur 2.1 op pagina 11 weergegeven beslisboom.



Figuur 2.2 Recursief gepartitioneerde kenmerkruimte.

De constructieprocedure omvat de drie stappen van: (1) het splitsen van knooppunten; (2) het bepalen welke knopen eindknopen zijn; en (3) het toekennen van klasselabels aan eindknopen.

De toekenning van klasselabels aan eindknopen is eenvoudig. De labels worden toegekend op grond van het aantal objecten per klasse dat zich in een betreffende eindknoop bevindt. Veelal is een meerderheid van objecten van één klasse bepalend voor het klasselabel, maar soms ook wordt het klasselabel aan de hand van een gewogen som van het aantal objecten per klasse bepaald; bijvoorbeeld wanneer bepaalde klassen waarschijnlijker zijn dan andere, worden de *a priori* waarschijnlijkheden als weegfactoren gebruikt.

De overgrote meerderheid van de literatuur over het construeren van beslisbomen heeft betrekking op de keuze van splitscriteria en het bepalen van eindknopen. In [Breiman *et al.*, 1984; Buntine & Niblett, 1992] wordt aangegeven dat de prestaties van een beslisboom niet significant variëren over een breed scala van 'aanvaardbare' splitscriteria, maar dat het werkelijke probleem neerkomt op het bepalen van de eindknopen. Het is immers niet moeilijk een boom zodanig te laten 'groeien', dat wanneer de leerverzame-

ling adequaat is (d.w.z. er geen identieke kenmerkpatronen met verschillend klasselabel zijn), de resubstitutiefout van de volledig uitgebreide boom gelijk aan nul is. Zoals is aangegeven sub 2.2.2, geeft de resubstitutieschatting in het algemeen geen goede indicatie van de werkelijke prestaties van een classificator op nieuwe objecten. Kenmerkend voor het inductieproces is, dat naarmate het aantal eindknopen toeneemt de resubstitutiefout afneemt, terwijl de werkelijke foutfractie eerst afneemt en vervolgens weer toeneemt omdat de boom overgespecialiseerd raakt. Hieruit volgt dat het groeiproces van de boom getermineerd moet worden zodra de werkelijke foutfractie zijn minimale waarde aanneemt. De resulterende boom is dan een subboom van de volledig uitgebreide boom.

Er bestaan verschillende benaderingen om de juiste (sub)boom te vinden die de beste prestaties levert. Eén daarvan is tijdens het splitsproces te bepalen of van verder splitsen van een knoop een significante bijdrage ter verbetering van de prestaties verwacht mag worden; zo niet, dan wordt de betreffende knoop tot eindknoop gemaakt. Significantie kan door middel van een standaard statistische test, bijvoorbeeld de Chi-kwadraat test, worden gemeten [Talmon, 1986]. Een andere benadering is eerst een uitgebreide boom te induceren, en deze vervolgens met behulp van *pruning*technieken te 'snoeien' door beslisknopen zodanig te verwijderen, dat de meest gunstige subboom van de aanvankelijke boom wordt gevonden. Voor een overzicht van enkele veelgebruikte pruningtechnieken wordt verwezen naar [Mingers, 1989a].

Het vinden van de 'juiste' boom zou echter pas van belang zijn als op een andere prestatiemaat dan de foutfractie kan worden geoptimaliseerd. In de volgende paragraaf zal daarom eerst nader worden ingegaan op het splitsen van knopen, waarbij de vraag zal worden beantwoord of en hoe splitscriteria kunnen worden aangewend om een inductiemethode te ontwikkelen die aan de in de inleiding geformuleerde doelstelling voldoet. Het bepalen van de 'juiste' uitwisseling tussen complexiteit en prestaties van een resulterende classificator wordt voorshands achterwege gelaten.

Het splitsen van knopen

Eerder is genoemd dat een boom door recursieve partitionering van de kenmerkruimte wordt geconstrueerd. Voor elke knoop dient te worden bepaald of en hoe deze moet worden gesplitst. De meest gevolgde benadering is de mogelijke kenmerken stuksgewijs te selecteren en voor verschillende drempelwaarden te evalueren hoe de prestaties van de boom erdoor worden beïnvloed. De evaluatie vindt doorgaans plaats met behulp van een zogenaamde splitsevaluatiefunctie.

Het doel van een splitsevaluatiefunctie is een kenmerk en bijbehorende drempelwaarde te selecteren waarmee een maximale verbetering van de prestaties van de boom wordt verkregen. De evaluatiefunctie baseert zich daarbij echter alleen op de splitsing die in één knoop wordt aangebracht, zonder daarbij mogelijke splitsingen van dieper- of hogergelegen knopen in beschouwing te nemen.

Onzuiverheidsmaten

De meest gebruikte splitsevaluatiefuncties geven een maat voor de 'onzuiverheid' van de verzameling objecten in een knoop. In [Breiman *et al.*, 1984, pagina 32] wordt aangetoond dat de functie ϕ een geschikte onzuiverheidsfunctie is wanneer: (*i*) ϕ maximaal is wanneer de klasse-conditionele waarschijnlijkheden (fracties) gelijk zijn, (*ii*) ϕ zijn minimum aanneemt dan en slechts dan als alle objecten in een knoop tot dezelfde klasse behoren, en (*iii*) ϕ een symmetrische functie in de klasse-conditionele waarschijnlijkheden (fracties) is.

Indien voor een binaire boom, gegeven een onzuiverheidsfunctie ϕ, de onzuiverheidsmaat $i(t)$ van een knoop t gedefinieerd wordt door

$$i(t) = \phi\left(p(1|t), p(2|t), \ldots, p(J|t)\right), \tag{2.3}$$

waarin $p(j|t)$, $1 \leq j \leq J$, de fractie van klasse-j objecten in knoop t is, dan is na splitsing van knoop t in een linker- en rechterknoop, respectievelijk aangeduid met t_L en t_R, de onzuiverheidsreductie

$$\Delta i(t) = i(t) - p_L i(t_L) - p_R i(t_R), \tag{2.4}$$

waarin p_L en p_R de respectievelijke fracties van het aantal objecten van knoop t dat zich na splitsing daarvan in de knoop t_L en t_R bevindt.

Enkele veelgebruikte onzuiverheidsmaten $i(t)$ zijn:

- de uit de klassieke informatietheorie bekende entropiemaat, $-\sum_j p(j|t)\log\left(p(j|t)\right)$, [Gleser & Collen, 1972; Sethi & Sarvarayudu, 1982; Breiman *et al.*, 1984; Quinlan, 1986; Talmon, 1986];
- de Gini-index, $1 - \sum_j p^2(j|t)$, [Breiman *et al.*, 1984]; en
- het Twoing-criterium, $\frac{p_L p_R}{4}\left[\sum_j |p(j|t_L) - p(j|t_R)|\right]^2$, [Breiman *et al.*, 1984].

Andere splitsevaluatiefuncties, zoals de Kolmogorov-Smirnoff afstand [Friedman, 1977; Rounds, 1980], welke een maat is voor de separabiliteit van twee univariate kansverdelingsfuncties, de traditionele Chi-kwadraat statistic [Hart, 1985] als maat voor de associatie tussen twee stochastische variabelen en de G statistic [Sokal & Rohlf, 1981; Talmon, 1986], die equivalent is met de uit de informatietheorie bekende Kullback informatiemaat, worden hier verder buiten beschouwing gelaten, omdat hun vermogen tot het reduceren van onzuiverheid een nodeloos ingewikkelde uiteenzetting vergt.

Van de onzuiverheidsmaten zal worden nagegaan in hoeverre ze geschikt of aan te passen zijn, zodat ze voor de specifieke toepassing binnen de medische diagnostiek kunnen worden gebruikt.

Uitgaande van het voorgaande, ligt het voor het minimaliseren van de foutfractie van de boom voor de hand de resubstitutiefout zèlf als onzuiverheidsmaat te nemen. Dit is equivalent met $i(t)$ gelijk te stellen aan $r(t)$, waarbij

$$r(t) = 1 - \max_j p(j|t)$$

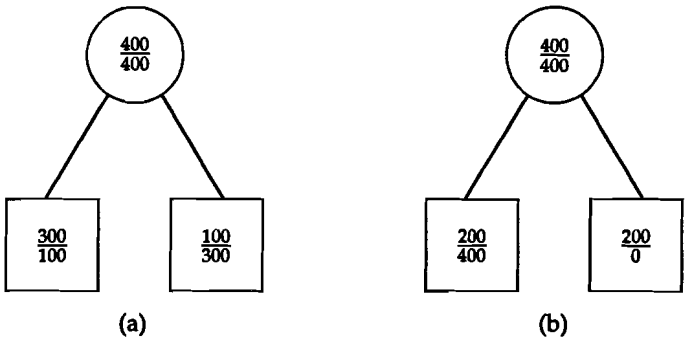
de resubstitutieschatting van de foutfractie in knoop t is. De overeenkomstige onzuiverheidsfunctie ϕ wordt gegeven door

$$\phi(p_1, ..., p_J) = 1 - \max_j p_j$$

en voldoet aan de genoemde voorwaarden (*i*), (*ii*) en (*iii*). Deze benadering kent echter een tweetal bezwaren.

Ten eerste kan de onzuiverheidsreductie nul zijn. In een tweeklassenprobleem waarbij beide klassen *a priori* even waarschijnlijk zijn en een meerderheid van de objecten in een knoop t van klasse j is, is het denkbaar dat elke splitsing van de knoop t, in knopen t_L en t_R resulteert die beide op zich eveneens een meerderheid aan klasse-j objecten bevatten. Aangetoond kan worden dat in dit specifieke geval de onzuiverheid van knoop t niet gereduceerd kan worden en het splitsen daarvan derhalve niet zinvol is.

Ten tweede kan de situatie zich voordoen waarbij twee verschillende splitsingen van een knoop dezelfde reductie in foutfractie opleveren, terwijl één van beide splitsingen vanuit strategisch oogpunt gezien de voorkeur geniet. Beschouw daartoe het in figuur 2.3 weergegeven voorbeeld van een tweeklassenprobleem waarbij beide klassen even waarschijnlijk zijn; van beide klassen zijn 400 objecten aanwezig. Zowel bij de



Figuur 2.3 Splitsingen met gelijke foutfracties volgens verschillende strategieën.

splitsing volgens (a) als die volgens (b) heeft de boom een foutfractie van 0.25, terwijl voor het verdere splitsproces de splitsing volgens (b) de voorkeur geniet, omdat de rechterknoop een zuivere eindknoop is die 200 objecten van dezelfde klasse bevat. In een volgende splitsstap zal alleen de linkerknoop worden gesplitst, terwijl bij de splitsing volgens (a) in volgende splitsstappen zowel de linker- als de rechterknoop zal worden

gesplitst. Binnen de context van het voortgezette 'groeiproces' waaraan de boom onderhevig is, blijkt de foutfractie als onzuiverheidsmaat dus geen wenselijk criterium te zijn. Dit wordt in hoofdzaak veroorzaakt door het feit dat het groeiproces een eenstaps-optimaliseringsprocedure is.

Bovenbeschreven probleem is te ondervangen door onzuiverheidsfuncties te kiezen die convex zijn (d.w.z. $\phi'' > 0$) [Breiman *et al.*, 1984, pagina 99]. De entropie, Gini-index en daarvan afgeleide onzuiverheidsmaten voldoen aan dit criterium.

Bezwaren tegen het gebruik van traditionele onzuiverheidsmaten

Empirische resultaten tonen aan dat onzuiverheidsmaten zeer geschikt zijn om beslisbomen te induceren, wanneer getracht wordt een classificator te construeren die de foutfractie minimaliseert [Mingers, 1989b]. Echter, het verband tussen de onzuiverheidsmaat en de te optimaliseren foutfractie is analytisch niet eenvoudig te bepalen. In [Sethi & Sarvarayudu, 1982] wordt een relatie tussen de entropie, de daarvan afgeleide mutuele informatie, en de foutfractie afgeleid die wordt gebruikt om het inductieproces zodanig te sturen dat de foutfractie van de resulterende boom kleiner is dan een tevoren opgegeven waarde. Helaas wordt in hun afleiding als schatting van de werkelijke foutfractie de resubstitutiefout gebruikt, en deze kan, zoals sub 2.2.2 is aangegeven, willekeurig klein worden gemaakt.

Een eerste bezwaar tegen het gebruik van de genoemde onzuiverheidsmaten is dat de relatie tussen splits- en prestatiemaat in het algemeen onduidelijk is, speciaal voor de diagnostische prestatiematen sensitiviteit en specificiteit, omdat deze zich niet zonder meer in termen van onzuiverheidsmaten laten uitdrukken. Directe toepassing van één van de gewenste prestatiematen als splitsmaat is onmogelijk, omdat eenvoudig is na te gaan dat niet aan de eerder genoemde voorwaarden is voldaan.

Een tweede bezwaar betreft de optimaliteit van de resulterende classificator. Zelfs al zou bereikt kunnen worden dat op grond van de hiervoor beschreven techniek een beslisboom is te induceren die aan de tevoren gedefinieerde prestatierestricties voldoet, dan nog kan niet worden gegarandeerd dat daadwerkelijk de beste boom of een benadering daarvan is gevonden. Dit bezwaar geldt ongeacht de splitsmaat, omdat het inductieproces een heuristisch eenstaps-optimaliseringsproces is.

Uit deze laatste constatering blijkt de noodzaak om te zoeken naar technieken waarbij beslissingen niet uitsluitend op grond van onvolledige, lokale informatie worden genomen, maar waarbij getracht wordt om door combinatie van lokale informatie op een meer globaal niveau beslissingen te nemen.

2.4 INDUCTIE VAN BESLISREGELS

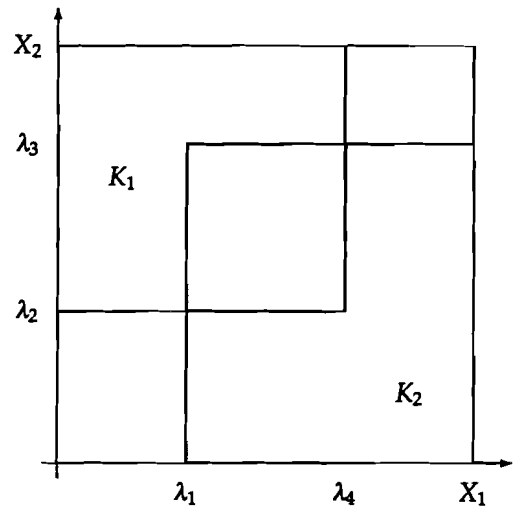
Zoals reeds in de vorige paragraaf is opgemerkt, kan met behulp van beslisregels, ook wel produktieregels genoemd, op een voor een expert begrijpelijke en inzichtelijke manier worden geclassificeerd. Een beslisregel legt een verband tussen kenmerken en kenmerkwaarden enerzijds en één bepaalde klasse anderzijds. Doorgaans heeft zij de vorm:

als <expr> **dan** <klasse>,

waarbij een klasselabel <klasse> één van de te onderscheiden klassen is en een expressie <expr> bestaat uit één of meer door logische operatoren (EN, OF) gecombineerde kenmerktests. Een kenmerktest heeft hierbij de volgende gedaante:

(<kenmerk> <operator> <drempelwaarde>).

Het is gebruikelijk om beslisregels in DNF-vorm (*disjunctive normal form*) weer te geven; dat wil zeggen als disjunctie van conjuncties. Evenals bij beslisbomen, is de kenmerkruimte



Figuur 2.4 Gepartitioneerde kenmerkruimte.

ook met behulp van beslisregels in verschillende subgebieden in te delen, waarbij aan een object dat binnen een subgebied valt, een klasselabel wordt toegewezen. Anders hierbij is, dat waar bij beslisbomen de equivalente verzameling beslisregels wederzijds uitsluitend is vanwege het recursief partitioneren van de kenmerkruimte, bij beslisregels dit geenszins het geval hoeft te zijn. Zo vormen de regels (2.5) en (2.6) bijvoorbeeld een mogelijke verzameling beslisregels,

als $(X_1 \leq \lambda_4)$ EN $(X_2 > \lambda_2)$ **dan** K_1 (2.5)

als $(X_1 > \lambda_1)$ EN $(X_2 \leq \lambda_3)$ **dan** K_2, (2.6)

waarvan de corresponderende indeling van de tweedimensionale kenmerkruimte in figuur 2.4 is weergegeven. Uit deze figuur blijkt dat twee verschillende subgebieden elkaar kunnen overlappen. Bovendien blijken bepaalde gedeelten van de kenmerkruimte niet door een klasselabel te zijn gekenmerkt.

Omdat bij praktische classificatietaken in het algemeen een eenduidig beslisuitspraak gewenst is, is zowel een beslismechanisme nodig dat aan een object dat aan meer dan één beslisregel voldoet, één definitief klasselabel toewijst, als een beslismechanisme dat een klasselabel toewijst aan niet-gelabelde subgebieden van de kenmerkruimte. Slechts op deze wijze wordt altijd een beslisuitspraak afgedwongen. Later volgt een voorbeeld van dergelijke, veelal op heuristische overwegingen gefundeerde beslismechanismen.

In de literatuur worden twee verschillende strategieën voor het induceren van beslisregels beschreven. In het hiernavolgende worden zij functioneel en beknopt beschreven. Tevens wordt nagegaan of deze methoden geschikt zijn voor de specifieke toepassing binnen de medische diagnostiek.

2.4.1 Generatie van produktieregels uit beslisbomen

Volgens een eerste strategie wordt een beslisboom geïnduceerd volgens de in paragraaf 2.3 beschreven methode. De resulterende boom wordt daarna als een verzameling conjunctieve beslisregels herschreven, door voor elke eindknoop in de boom de kenmerktests op het pad van de wortel naar de betreffende eindknoop door middel van de logische EN-operator te combineren. Vervolgens wordt de verzameling regels gepruned door voor elk van de kenmerktests in de conjunctieve regels na te gaan in welke mate een test relevant is voor de klasse die met de regel correspondeert. Dit wordt per regel in een *certainty factor* tot uitdrukking gebracht. Een kenmerktest wordt uit de regel verwijderd indien zijn verwijdering niet tot een verslechtering van de resubstitutienauwkeurigheid van de regel leidt, of de test vanwege overspecialisatie op het leerbestand een hoge schijnbare nauwkeurigheid heeft, maar statistisch gezien irrelevant blijkt te zijn. De eigenschap dat de van een beslisboom afgeleide verzameling beslisregels wederzijds uitsluitend is, gaat hiermee verloren.

Het classificeren van een object verloopt als volgt. Eerst worden die regels gevonden die 'vuren' op het object, dat wil zeggen regels die aan alle testcondities beantwoorden. Indien er meer dan één regel is die vuurt, dan wordt de regel met de hoogste *certainty factor* gekozen. Is er geen regel die vuurt, dan wordt de verstekklasse gekozen, hetgeen doorgaans de meest voorkomende klasse in het leerbestand is. Een typisch voorbeeld van deze techniek is beschreven in [Quinlan, 1987a; Quinlan, 1987b].

De strategie om beslisregels uit beslisbomen te genereren in plaats van de regels op een directe wijze uit het leerbestand af te leiden heeft tenminste twee voordelen. Ten eerste hebben praktische classificatietaken veelal kenmerken met een continu waardenbereik, waarvoor geschikte drempelwaarden bepaald dienen te worden. De hiërar-

chische benadering van recursieve partitionering verschaft hiervoor een krachtige en contextgevoelige manier. Ten tweede is, doordat het langste pad in een beslisboom doorgaans minder kenmerktests omvat dan er verschillende kenmerken zijn, de omvang van de zoekruimte drastisch verkleind. Wanneer het leerbestand bijvoorbeeld objecten bevat die door n kenmerken worden beschreven en geen enkel pad in de beslisboom meer dan k kenmerktests omvat, dan is de oplossingsruimte van potentiële beslisregels derhalve van orde $O(2^n)$ tot $O(2^k)$ teruggebracht.

Een nadeel van deze methode in het licht van onze doelstelling is, dat als optimaliseringscriterium wederom nauwkeurigheid is gekozen. Andere prestatiematen komen niet aan de orde en laten zich, evenals bij beslisbomen, tijdens het inductieproces niet eenduidig beïnvloeden. Het is daarom niet duidelijk op welke wijze deze methode kan bijdragen aan de oplossing van het probleem.

2.4.2 Inductie van de enkele 'beste' regel

Volgens een tweede strategie wordt voor één klasse één enkele 'beste' regel gevonden, waarna de objecten die door die regel worden bestreken, uit het leerbestand worden verwijderd. Dit wordt herhaald totdat geen enkel object resteert. Voorbeelden hiervan zijn het AQ- en CN-algoritme [Michalski *et al.*, 1986; Clark & Niblett, 1989] en PVM [Weiss *et al.*, 1987; Weiss *et al.*, 1990]. In het navolgende worden deze algoritmen meer in detail besproken, met als doel na te gaan of zij aan de in de inleiding geformuleerde doelstelling voldoen of kunnen worden aangepast.

AQ15 en CN2

Het classificatieformalisme van AQ en CN2 is gebaseerd op de in paragraaf 2.4 geïntroduceerde vorm van beslisregels. Gezegd wordt dat een expressie of regel een object 'bedekt' als zij 'waar' is voor dat object. De klasse die met een expressie wordt geassocieerd is de meest voorkomende klasse van objecten in het leerbestand die door de expressie worden bedekt.

Een nieuw object wordt geclassificeerd door te zoeken naar die geïnduceerde regels waarbij het object aan alle testcondities daarvan voldoet. Indien het object aan slechts één enkele regel voldoet, dan wordt de klasse die met de betreffende regel is geassocieerd aan het object toegewezen. Voldoet het object daarentegen aan meerdere regels, dan wordt het aantal objecten in elke bedekte klasse bepaald en wordt de klasse met de hoogste frequentie toegewezen. Indien het object tenslotte door geen enkele regel wordt bedekt, dan wordt zij bij verstek aan die klasse die het vaakst in het leerbestand voorkwam toegewezen. Op deze wijze wordt de gehele beslisruimte in afzonderlijke gebieden ingedeeld.

Het leeralgoritme van AQ en CN2 genereert achtereenvolgens voor elke klasse één enkele 'beste' regel. Bij een gegeven klasse wordt een disjunctie van conjuncties gevormd

om als het conditionele gedeelte van een regel voor die klasse te fungeren. Dit proces verloopt in stappen; in elke stap wordt op grond van een eerste heuristiek een verzameling van 'veelbelovende' conjuncties gegenereerd die een willekeurig pósitief object (de *seed*), maar geen enkel négatief object van het leerbestand bedekt; wordt uit deze verzameling middels een tweede heuristiek de 'beste' conjunctie gekozen; en worden de objecten die door deze conjunctie worden bedekt uit het leerbestand verwijderd. Deze stap wordt herhaald totdat voldoende conjuncties zijn gevonden om bij disjunctieve combinatie alle objecten van de gekozen klasse te bedekken. Het gehele proces wordt achtereenvolgens voor elke klasse uitgevoerd.

De eerste heuristiek die AQ gebruikt om een verzameling veelbelovende conjuncties te genereren is de som van het aantal positieve voorbeeldobjecten dat bedekt wordt en het aantal negatieve voorbeeldobjecten dat wordt uitgesloten, te maximaliseren. Als tweede heuristiek wordt het aantal positieve voorbeeldobjecten dat bedekt wordt gemaximaliseerd, om uit deze verzameling de beste conjunctie te kiezen.

Bij het genereren van een conjunctie voert het AQ-algoritme een algemeen-naar-specifiek zoekprocedure uit voor het vinden van een veelbelovende conjunctie. Er worden echter alleen specialisaties beschouwd die een bepaald bedekt negatief voorbeeldobject van de conjunctie uitsluiten, terwijl er voor wordt gezorgd dat de *seed* bedekt blijft, en er wordt geïtereerd totdat alle negatieve voorbeeldobjecten zijn uitgesloten. Als resultaat hiervan doorzoekt AQ alleen alle mogelijke conjuncties die consistent zijn met het leerbestand.

Het leeralgoritme van CN2 is in hoofdzaak gelijk aan dat van AQ, zij het dat de zoekprocedure zodanig is gewijzigd dat zij onafhankelijk is van specifieke voorbeeldobjecten, en daardoor regels kan genereren die niet perfect presteren op het leerbestand. Dit is bewerkstelligd door het specialisatieproces te verbreden en alle mogelijke specialisaties van een conjunctie te evalueren, op een wijze die te vergelijken is met die van het evalueren van alle kenmerktests in een beslisknoop tijdens het induceren van een beslisboom. Gezocht wordt naar een conjunctie die een zo groot mogelijk aantal voorbeeldobjecten van één klasse bedekt, en daarbij een zo klein mogelijk aantal van de andere klassen bedekt. De conjunctie moet daarbij zowel 'voorspellend' (d.w.z. een groot aantal objecten van de gekozen klasse wordt bedekt en weinig van de andere klassen) als 'betrouwbaar' (d.w.z. niet bij toeval voortgekomen uit het leerbestand) zijn. Dit wordt bereikt met behulp van hierna te noemen evaluatiefuncties. Een conjunctie die hieraan voldoet, wordt aan een beperkte verzameling van oplossingen toegevoegd. Dit proces itereert totdat er geen conjuncties meer gevonden worden. Vervolgens worden de in de verzameling opgenomen conjuncties op alle mogelijke manieren verder gespecialiseerd, en wordt geëvalueerd welke nieuwe oplossingen vooruitzichten bieden. Een evaluatiefunctie beperkt in deze stap de omvang van de nieuwe verzameling door de laagst gewaardeerde elementen te verwijderen.

Voor het bepalen van de kwaliteit van conjuncties, maakt het CN2-algoritme als evaluatiefunctie gebruik van de uit de informatietheorie afkomstige entropiemaat. Deze maat geeft de voorkeur aan conjuncties die een groot aantal voorbeeldobjecten van één klasse bedekken en daarbij weinig van de andere klassen. Met een statistische evaluatiefunctie als de Chi-kwadraat test evalueert het algoritme of een conjunctie significant is en of er daarmee een correlatie bestaat tussen kenmerken en kenmerkwaarden enerzijds en klassen anderzijds. Bij beslisbomen is een dergelijke benadering al eerder aan de orde gesteld [Talmon, 1986].

Evenals bij het induceren van beslisbomen wordt bij deze techniek de nauwkeurigheid gemaximaliseerd en komen andere prestatiematen niet aan de orde. Bovendien is het niet mogelijk om de prestaties van de resulterende classificator te beïnvloeden en is de relatie tussen de heuristieken (evaluatiefuncties) en de te optimaliseren prestatiemaat onduidelijk, met name voor de diagnostische prestatiematen sensitiviteit en specificiteit. De algoritmen verschaffen daarnaast geen inzicht in de mate van optimaliteit van de resulterende oplossing. Hieruit wordt geconcludeerd dat de wijze waarop deze algoritmen zouden kunnen worden aangepast om daarmee aan de doelstelling te voldoen, onduidelijk is.

PVM

Het PVM-algoritme genereert DNF-regels door afzonderlijke kenmerktests door middel van logische EN- en OF-operatoren met elkaar te combineren, en is daarbij in staat om één van de vijf diagnostische prestatiematen te maximaliseren onder oplegging van minimumwaarden aan de resterende prestatiematen. Een extra restrictie hierbij is dat het aantal kenmerktests dat in een beslisregel wordt toegestaan een vooraf opgegeven waarde niet overschrijdt.

Dit algoritme heeft een aantal voordelen boven andere inductietechnieken. Ten eerste is het mogelijk de prestaties van de resulterende classificator op een eenduidige wijze te beïnvloeden. Ten tweede kan als optimaliseringscriterium een diagnostische prestatiemaat worden gekozen. Tenslotte worden kenmerkcombinaties uitvoeriger geevalueerd dan bij andere algoritmen, hetgeen de optimaliteit ten goede komt. PVM benadert het inductieprobleem met een zoekprocedure die de enkele 'beste' regel tracht te vinden waarvan de lengte kleiner of gelijk is aan een vooraf opgegeven waarde. In tegenstelling tot technieken voor het induceren van beslisbomen, waarbij een classificator op grond van lokale optimaliseringsstappen wordt geconstrueerd, begeeft deze methode zich op een meer globaal niveau en is het een heuristische benadering voor het uitputtend genereren van alle mogelijke beslisregels van een zekere lengte. Alhoewel een uitputtende zoektocht door de oplossingsruimte van alle regels in DNF-vorm voor praktische toepassingen vaak onuitvoerbaar is vanwege de combinatorische explosie die optreedt, blijkt de zoekruimte met een klein aantal heuristieken tot aanvaardbare proporties te kunnen worden teruggebracht.

Een benadering voor uitputtend zoeken door de oplossingsruimte van alle DNF-regels van lengte l bestaat uit het genereren van alle mogelijke Booleaanse expressies van l kenmerkvariabelen en $l-1$ operatoren (EN, OF) zonder daarbij logisch equivalente expressies te genereren, en bij elke nieuwe expressie te testen of deze beter is dan de tot dan toe gevonden oplossing. Het leeralgoritme van PVM is gebaseerd op een *beam search* strategie, waarbij een door heuristieken beperkte verzameling van de meest veelbelovende expressies wordt bijgehouden en combinaties van deze expressies op hun beurt gebruikt worden om langere expressies te genereren. De meest veelbelovende langere expressies worden op hun beurt weer gebruikt om nog langere expressies te genereren. Een voorbeeld van de resulterende verzameling expressies die ontstaat wanneer voor kenmerkvariabelen A, B en C expressies tot en met lengte 2 worden gegenereerd, is in tabel 2.4 weergegeven.

Tabel 2.4 Voorbeeld van gegenereerde expressies.

A
B
C
A EN B
A EN C
B EN C
A OF B
A OF C
B OF C

Voor elke kenmerkvariabele wordt voorafgaand aan het genereren van expressies een rekenkundige operator ($\leq$, $>$) en een verzameling drempelwaarden bepaald. Als drempelwaarden worden die waarden gekozen waarbij de positief predictieve waarde een lokaal maximum aanneemt. De kenmerkvariabelen van de expressies worden vervolgens op alle mogelijke manieren met drempelwaarden geïnstantieerd en er wordt bepaald welke regels goed presteren. Als, voortbordurend op het voorgaande voorbeeld, kenmerk X_A en X_B respectievelijke drempelwaarden λ_1^A, λ_2^A en $\lambda_1^B, \lambda_2^B, \lambda_3^B$ hebben, dan geeft de expressie A EN B bij instantiatie aanleiding tot de in tabel 2.5 op pagina 23 weergegeven mogelijkheden. Omdat nieuwe, langere expressies door korte expressies worden voortgebracht, komen expressies die eenmaal door heuristieken uit de verzameling zijn verwijderd, nooit meer in langere expressies voor.

Tijdens het instantiëren van de kenmerkvariabelen worden heuristieken toegepast om het aantal mogelijkheden te beperken. Zo worden bij het optimaliseren op predictieve waarde of nauwkeurigheid expressies waarvan de sensitiviteit- en specificiteitswaarde

Tabel 2.5 Voorbeeld van geïnstantieerde expressies.

$(X_A \leq \lambda_1^A)$ EN $(X_B \leq \lambda_1^B)$
$(X_A \leq \lambda_1^A)$ EN $(X_B \leq \lambda_2^B)$
$(X_A \leq \lambda_1^A)$ EN $(X_B \leq \lambda_3^B)$
$(X_A \leq \lambda_2^A)$ EN $(X_B \leq \lambda_1^B)$
$(X_A \leq \lambda_2^A)$ EN $(X_B \leq \lambda_2^B)$
$(X_A \leq \lambda_2^A)$ EN $(X_B \leq \lambda_3^B)$

beide kleiner zijn dan de opgegeven minimumwaarden direct geëlimineerd, omdat verwacht wordt dat zij niet tot bruikbare regels zullen bijdragen. Een expressie met een lagere specificiteitswaarde dan opgegeven, wordt ook bij het op sensitiviteit optimaliseren geëlimineerd, omdat wanneer deze expressie door middel van een OF-operator met een andere expressie wordt gecombineerd de specificiteitswaarde afneemt. Nadat alle interessante regels zijn gegenereerd, wordt de beste regel als antwoord gepresenteerd.

Het PVM-algoritme is in principe ontworpen voor een tweeklassenprobleem. Voor meerklassenproblemen kan PVM worden toegepast door elke klasse afzonderlijk tegen de resterende klassen af te zetten. Een N-klassenprobleem wordt hierdoor als N tweeklassenproblemen beschouwd. Nadat alle N beslisregels zijn gevonden, zouden bijvoorbeeld de beste $N-1$ regels kunnen worden gebruikt, terwijl de resterende klasse als verstekklasse wordt gekozen. Een andere mogelijkheid is uit de N beslisregels één beste regel te kiezen, vervolgens alle objecten die door die regel worden bedekt uit het leerbestand te verwijderen, en deze procedure recursief toe te passen. Hier wordt verder niet op ingegaan, omdat dit buiten het bestek van het onderzoek ligt.

Omdat PVM niet aangeeft welke regellengte optimaal is—bij toenemende regellengte geeft de resubstitutieschatting een steeds optimistischer resultaat, terwijl het gevaar voor overspecialisatie bestaat—dient die regellengte te worden bepaald waarvoor de te optimaliseren prestatiemaat maximaal is. PVM kan hiervoor bij uitstek met resamplingtechnieken worden gecombineerd. Het resamplingprincipe beperkt zich immers niet tot het schatten van prestatiematen alleen, maar kan worden gebruikt om elke willekeurige parameter van een populatie te schatten [Efron & Gong, 1983]. Voor elke door PVM gegenereerde regel van lengte l wordt door middel van de *leave-one-out* of *random resampling* techniek een schatting van de waarde van de te optimaliseren prestatiemaat bepaald. De optimale regellengte, k, is dan die lengte waarvoor de geschatte prestatiemaat maximaal is. Wanneer de optimale regellengte bepaald is, kan het PVM-algoritme opnieuw worden gebruikt om, op basis van alle voorbeeldobjecten, de uiteindelijk beste beslisregel van de optimale lengte te construeren.

2.5 CONCLUSIES

De PVM-benadering komt hiermee het dichtst in de buurt van de in de inleiding geformuleerde doelstelling. Alhoewel optimaliteit niet gegarandeerd is door de heuristische *beam search* strategie en het heuristisch bepalen van de drempelwaarden, is het de enige tot nu toe bekende methode die een combinatie van meerdere kenmerktests voor meerdere drempelwaardencombinaties evalueert. Een ander belangrijk voordeel is dat sturing van de prestatiematen mogelijk is en als optimaliseringscriterium één van de vijf diagnostische prestatiematen sensitiviteit, specificiteit, positief predictieve waarde, negatief predictieve waarde, en nauwkeurigheid kan worden gekozen. Een nadeel is echter dat het met name geschikt is voor relatief eenvoudige, laagdimensionale classificatietaken. Empirische resultaten hebben echter aangetoond dat een relatief korte regel vaak betere resultaten kan geven dan een verzameling meer uitgebreide regels [Michalski *et al.*, 1986; Holte, 1993]. Een benadering in deze richting lijkt derhalve vooruitzichten te bieden. Omdat er hiermee, evenals bij het induceren van beslisbomen het geval is, duidelijke aanwijzingen bestaan dat een benadering waarbij meer uitgebreid of zelfs uitputtend wordt gezocht de enige oplossing lijkt te zijn, zal in het volgende hoofdstuk nader worden ingegaan op methoden waarmee classificatoren uitputtend kunnen worden doorzocht.

HOOFDSTUK 3

Ontwikkeling van een nieuw algoritme

3.1 INLEIDING

Zoals in het vorige hoofdstuk is aangegeven, zijn conventionele inductietechnieken voor toepassing binnen de geneeskunde behept met twee nadelen. Ten eerste zijn de diagnostische prestatiematen van de resulterende classificator moeilijk te sturen. Ten tweede is globale optimaliteit van de resulterende classificator niet gegarandeerd.

Omdat meer uitgebreide zoektechnieken deze nadelen kunnen ondervangen, wordt in dit hoofdstuk een methode gepresenteerd die alle mogelijke classificatoren—beslisbomen of -regels—van een zekere complexiteit genereert. Tevens wordt daarbij het eerst genoemde nadeel ondervangen. Daartoe wordt uitgegaan van een methode waarbij de oplossingsruimte uitputtend wordt doorzocht en telkens wordt geverifieerd of de prestaties van de gevonden classificatoren voldoen aan de voorwaarden die vooraf zijn opgelegd. De rekenkundige complexiteit van de zoekmethode wordt geanalyseerd om inzicht te verkrijgen in de mate waarin de verschillende parameters effect hebben op de omvang van de oplossingsruimte. In aansluiting hierop worden een aantal heuristieken geïntroduceerd die deze omvang trachten te reduceren.

3.2 UITPUTTEND ZOEKEN IN BESLISBOMEN

Een eerste benadering bestaat er uit alle mogelijke bomen van een vooraf opgegeven complexiteit—meestal afgemeten aan het aantal beslisknopen— te genereren en de beste te selecteren, gegeven tevoren gestelde randcondities.

De ingrediënten voor een dergelijke benadering zijn, naast een bestand van voorbeeldobjecten, een algoritme dat op systematische wijze alle mogelijke classificatiebomen kan genereren, een mechanisme om de prestaties van die bomen te evalueren en te vergelijken, en een verzameling voorwaarden waaraan de prestaties van een gezochte classificator dienen te voldoen.

Voor wat betreft dit laatste aspect, beperken wij ons tot de vijf sub 2.2.1 genoemde diagnostische prestatiematen sensitiviteit, specificiteit, positief predictieve waarde, negatief predictieve waarde en nauwkeurigheid. Evenals bij het sub 2.4.2 besproken PVM-algoritme is het wenselijk om één van deze prestatiematen te maximaliseren onder restricties die aan de resterende maten zijn opgelegd.

Technieken om de prestaties van classificatoren te evalueren zijn in paragraaf 2.2 reeds aan de orde gesteld. Daar is onder meer geconcludeerd dat resamplingtechnieken bij uitstek geschikt zijn om betrouwbare schattingen van de werkelijke prestaties van een classificator te verschaffen. Omdat de complexiteit van 'de beste' classificator *a priori* onbekend is, dienen bomen van verschillende complexiteit te worden voortgebracht.

Indien voor bomen van verschillende complexiteit de bijbehorende, te maximaliseren, geschatte prestatiemaat tegen de complexiteit wordt uitgezet, wordt vaak een karakteristiek puntenverloop verkregen dat wordt gekenmerkt door een eerste gedeelte waarin de prestatiemaat relatief snel beter wordt, gevolgd door een tweede, hoofdzakelijk horizontaal gedeelte waarin de prestatiemaat langdurig nagenoeg constant is, en een derde gedeelte waarin de prestatiemaat geleidelijk slechter wordt. Binnen het tweede gedeelte zal de waarde van de prestatiemaat enigszins rond de maximumwaarde fluctueren. De positie van het maximum is mogelijkerwijs instabiel, omdat kleine veranderingen in parameterwaarden of in de samenstelling van de leer- en testpartities relatief grote veranderingen in de complexiteit van de bijbehorende beslisboom tot gevolg kunnen hebben. Om een uiteindelijke 'beste' boom te kiezen, wordt in [Breiman *et al.*, 1984, pagina's 78–80] gebruik gemaakt van een met *'1 SE rule'* aangeduide heuristiek. Deze heuristiek tracht de hiervoor aangegeven instabiliteit te reduceren en kiest de minst complexe beslisboom met een bijbehorende prestatiemaatwaarde die binnen een band van één *standard error* rond de maximumwaarde ligt.

Aanmerkelijk moeilijker is het een algoritme te ontwikkelen dat systematisch alle mogelijke beslisbomen kan genereren en dit zodanig efficiënt doet dat eerder geëvalueerde bomen buiten beschouwing kunnen worden gelaten. In het hiernavolgende wordt een mogelijke methode hiervoor meer in detail beschreven.

3.2.1 Een methode voor het genereren van beslisbomen

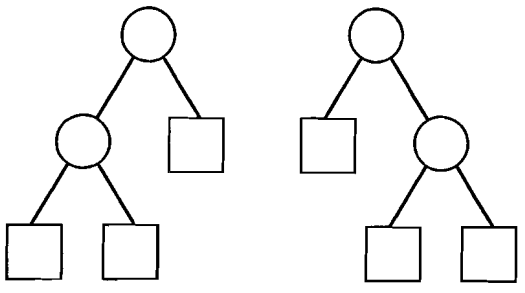
Een algoritme voor het genereren van alle mogelijke beslisbomen zal in staat moeten zijn om voor een gegeven klasse van bomen—te weten binaire bomen—alle mogelijke boomtopologieën voort te brengen en de beslis- en eindknopen daarvan op alle mogelijke manieren te voorzien van kenmerktests en klasselabels, zodat geïnstantieerde bomen worden verkregen. In het navolgende wordt een complexiteitsanalyse gepresenteerd van het genereren van boomtopologieën, het toekennen van klasselabels aan de eindknopen en het met kenmerktests invullen van de beslisknopen. Deze analyse geeft een uiteindelijke ordegrootte-indicatie van het uitputtend genereren van beslisbomen.

3.2.2 Complexiteitsanalyse

Om een indicatie van de complexiteit te krijgen, wordt verondersteld dat een voorbeeldobject door n kenmerkwaarden wordt beschreven en dat beslisbomen met k beslisknopen moeten worden gegenereerd. Voorts worden rekenkundige operatoren ($\leq, >$) toegestaan en wordt aangenomen dat elk kenmerk c potentiële drempelwaarden heeft.

Generatie van boomtopologieën

In een eerste stap van het generatieproces worden alle binaire bomen met k beslisknopen gegenereerd. Voor $k = 0$ en $k = 1$ is het duidelijk hoe de bijbehorende boomstructuren er uit zien, maar voor $k \geq 2$ bestaan er meerdere niet-equivalente boomtopologieën. Voor $k = 2$ bijvoorbeeld, zijn de twee in figuur 3.1 afgebeelde topologieën mogelijk.



Figuur 3.1 Boomtopologieën van binaire bomen met drie bladeren.

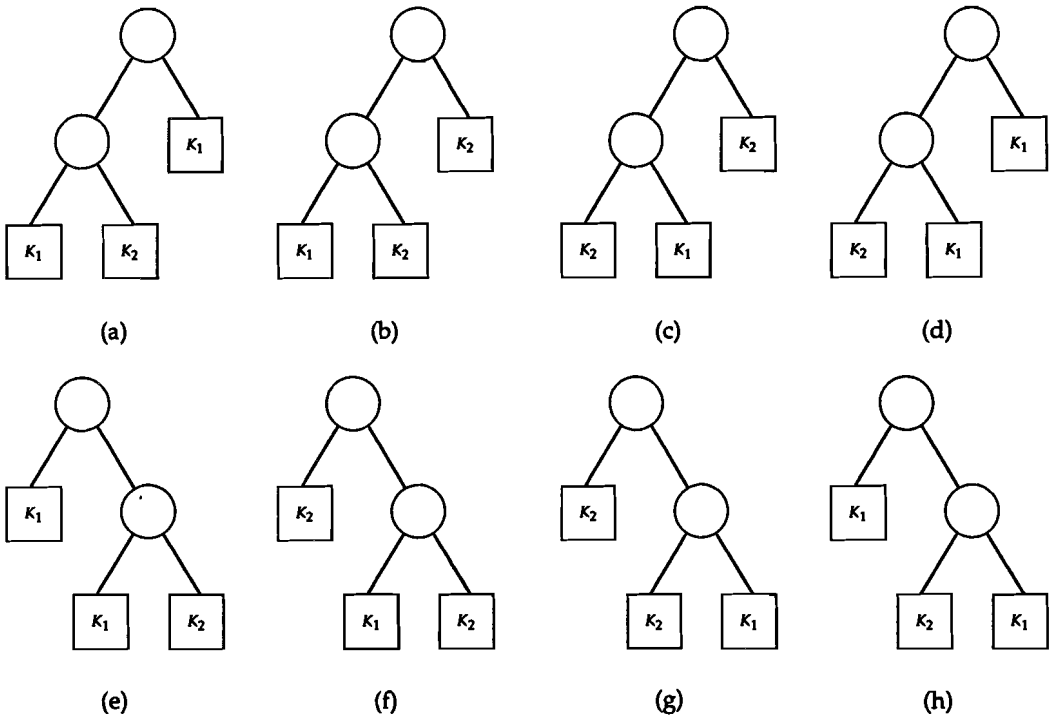
In bijlage A.1 is afgeleid dat het aantal verschillende topologieën van binaire bomen met k beslisknopen gelijk is aan

$$a_k = \frac{1}{k+1}\binom{2k}{k}, \qquad n \geq 0. \tag{3.1}$$

Voor $k = 1, 2, 3, 4, \ldots$ zijn er $1, 2, 5, 14, \ldots$ bomen.

Toekenning van klasselabels

In een tweede stap worden vervolgens op alle mogelijke manieren aan de $k+1$ eindknopen van de zojuist gegenereerde boomtopologieën één van twee mogelijke klasselabels (K_1, K_2) toegekend, met de restrictie dat aan twee eindknopen met dezelfde ouder als beslisknoop, een verschillend label wordt gegeven. Toewijzing van klasselabels aan de in figuur 3.1 afgebeelde topologieën vindt zodoende op de acht in figuur 3.2 weergegeven manieren plaats.



Figuur 3.2 Toewijzing van klasselabels aan boomtopologieën.

In bijlage A.2 is afgeleid dat het aantal manieren waarop aan de a_k niet-equivalente boomtopologieën van binaire bomen met k beslisknopen twee verschillende klasselabels kunnen worden toegewezen, aangeduid met b_k, tot uitdrukking kan worden gebracht door

$$b_k = \frac{2^{3k/2+1}}{2k+1}\left[P_{k+1}(\sqrt{2}) - P_{k-1}(\sqrt{2})\right], \qquad k \geq 0. \tag{3.2}$$

Hierin is $P_k(x)$ het Legendrepolynoom van de graad k. Voor $k = 1, 2, 3, 4, \ldots$ zijn er $b_k = 2, 8, 36, 176, \ldots$ gelabelde topologieën.

Instantiatie van beslisknopen

In een derde stap tenslotte, wordt de inhoud van de beslisknopen van de b_n bomen op alle mogelijke manieren ingevuld met een kenmerktest van de vorm

(<kenmerk> <operator> <drempelwaarde>).

Het aantal manieren om de k beslisknopen van een boom te instantiëren bedraagt derhalve[1]

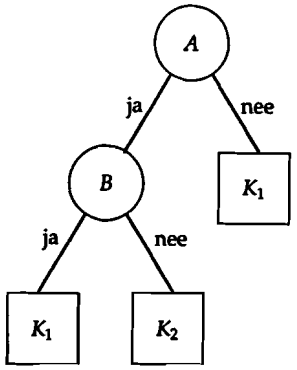
$$n^k \times 2^k \times c^k.$$

Het totaal aantal beslisbomen ligt hiermee in de orde van

$$n^k \times 2^k \times c^k \times b_k. \tag{3.3}$$

Opgemerkt dient te worden dat door het op deze wijze instantiëren lang niet alle bomen semantisch gezien zinvol zijn. Het instantiëren geeft daarnaast aanleiding tot redundante classificatoren. Dit wordt aan de hand van voorbeelden in het hiernavolgende toegelicht.

Door te veronderstellen dat een kenmerktest thans als testvariabele wordt aangeduid en met een hoofdletter wordt weergegeven, bijvoorbeeld $A = (X_A \leq \lambda_i^A)$ en $B = (X_B > \lambda_i^B)$, wordt vanuit een hoger gelegen standpunt bekeken welke classificatoren in de derde stap ontstaan.



Figuur 3.3 Logische instantiatie van een boom.

Wanneer de wortel van de boom van figuur 3.2(a) met testvariabele A wordt ingevuld en de andere beslisknoop met B, dan resulteert dat in een geïnstantieerde boom zoals die in figuur 3.3 is weergegeven. Hierbij is wederom afgesproken dat de linkertak wordt gevolgd als de uitkomst van de test in de betreffende beslisknoop positief uitvalt en dat anders de rechtertak wordt gevolgd. In paragraaf 2.3 is aangegeven hoe, uitgaande van

[1]Wanneer het aantal potentiële drempelwaarden per kenmerk c_i bedraagt, dan moet de term c^k worden vervangen door het k-voudige produkt van c_i voor alle in de beslisknopen ingevulde kenmerken X_i.

een geïnstantieerde beslisboom, logische beslisregels kunnen worden afgeleid. Zodoende worden in het voorbeeld van figuur 3.3 alle objecten die aan de expressie (A EN B) OF $\bar{A}$ voldoen door de classificator aan klasse K_1 toegewezen. De ontkenning van A wordt hierbij door $\bar{A}$ gerepresenteerd. Het is eenvoudig na te gaan dat deze expressie ook als $\bar{A}$ OF B is te schrijven. Op overeenkomstige wijze kan voor alle acht bomen in figuur 3.2 worden nagegaan dat, bij invulling van testvariabele A in de wortel en B in de eronder gelegen beslisknoop, de objecten aan klasse K_1 worden toegewezen wanneer zij voldoen aan de expressies zoals die in tabel 3.1 zijn weergegeven.

Tabel 3.1 Resulterende expressies.

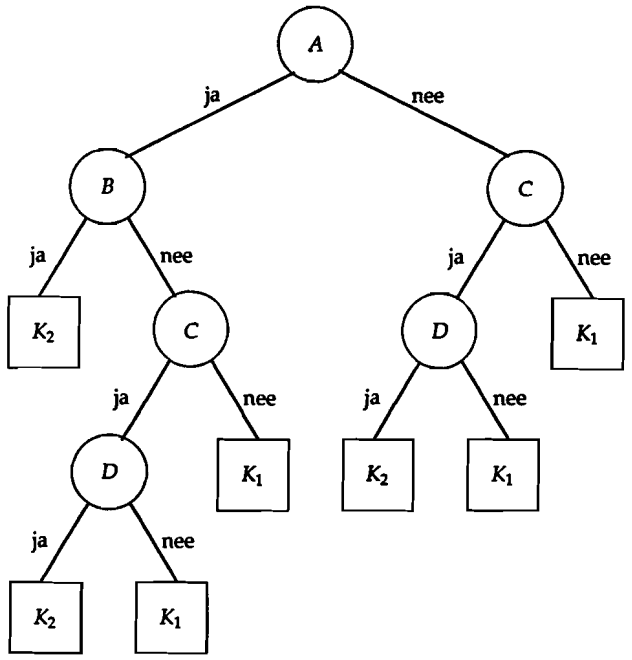
$E(a)$	=	$\bar{A}$ OF B	$E(e)$	=	A OF B
$E(b)$	=	A EN B	$E(f)$	=	$\bar{A}$ EN B
$E(c)$	=	A EN $\bar{B}$	$E(g)$	=	$\bar{A}$ EN $\bar{B}$
$E(d)$	=	$\bar{A}$ OF $\bar{B}$	$E(h)$	=	A OF $\bar{B}$

Uit tabel 3.1 blijkt dat er feitelijk niets anders plaatsvindt dan dat alle mogelijke Booleaanse expressies van twee logische testvariabelen A, B worden gegenereerd. De vraag die zich nu voordoet is waarom dergelijke expressies niet direct worden gegenereerd, in plaats van eerst alle boomtopologieën te genereren, vervolgens klasselabels toe te kennen, en tenslotte op alle mogelijke manieren de beslisknopen met kenmerktests in te vullen. Dit laatste is overigens niet evident wanneer geëist wordt dat de resulterende beslisbomen allen moeten verschillen. Indien immers in het hierboven beschreven voorbeeld van figuur 3.2 in een volgende stap de testvariabelen A en B zouden worden verwisseld, zodat B in de wortel wordt ingevuld en A in de andere beslisknoop, dan levert dat verschillende boominstantiaties op, terwijl de logische expressies die hieruit ontstaan allen reeds zijn voortgebracht. Worden bijvoorbeeld in de beslisknopen van de in figuur 3.2(b) weergegeven boomstructuur de testvariabelen A en B verwisseld, dan levert dat twee verschillende boominstantiaties op, terwijl zij logisch gezien dezelfde expressie opleveren; in het ene geval A EN B, in het andere geval B EN A (voor K_1). Door de commutatieve eigenschap van de logische operatoren EN en OF kunnen verschillende bomen (d.w.z. instantiaties daarvan) logisch gezien dezelfde expressie (en dus beslisregel) representeren. Uit het hierboven beschreven voorbeeld van bomen met 2 beslisknopen zou overigens ten onrechte de indruk kunnen zijn ontstaan dat herhaling van eerder gegenereerde expressies kan worden voorkomen door het permuteren van de testvariabelen achterwege te laten. Permutatie blijkt voor $k > 2$ echter noodzakelijk om ingewikkelder expressies te kunnen genereren.

Bezwaren tegen het uitputtend genereren van beslisbomen

Een benadering langs deze weg heeft dus tot gevolg dat ofwel onnodig veel bomen zouden worden gegenereerd die tot dezelfde beslisregel reduceren, ofwel een bijzonder uitgebreide administratie zou moeten worden bijgehouden van reeds gegenereerde expressies, waardoor het algoritme bijzonder ingewikkeld zou worden.

Een ander nadeel van het genereren van alle mogelijke bomen wordt gevormd door de impliciete eigenschap van beslisbomen dat ze wederzijds uitsluitende beslisregels opleveren. Zo kan het voorkomen dat een bepaalde subboom moet worden herhaald om aan deze eis te voldoen. In de literatuur is dit fenomeen bekend als het *'replication problem'* [Weiss & Kulikowski, 1990, pagina 132]. Uit het volgende voorbeeld blijkt dat beslisregels een doelmatiger representatieformalisme kunnen zijn dan beslisbomen. Om bijvoorbeeld voor klasse K_2 de relatief eenvoudige expressie (*A* EN *B*) OF (*C* EN *D*) in een beslisboom te representeren, is de in figuur 3.4 weergegeven boom nodig. Omwille van



Figuur 3.4 Boom voor expressie (*A* EN *B*) OF (*C* EN *D*).

de wederzijdse uitsluiting blijkt een subboom te moet worden herhaald. Een mogelijke oplossing voor dit probleem is Booleaanse expressies toe te staan in de beslisknopen. Het probleem om daarbij de beste combinatie van kenmerktests te vinden is, zoals sub 2.4.2 bij bespreking van het PVM-algoritme [Weiss *et al.*, 1987] ter sprake kwam, echter zeer complex en zal derhalve voor beslisbomen verder niet worden beschouwd.

Vanwege de bezwaren die de hiervoor beschreven beslisboom-benadering met zich meebrengt, en de voordelen die beslisregels in dit opzicht bieden, wordt het principe van beslisboomclassificatoren verlaten en wordt een mogelijke methode beschreven voor het uitputtend genereren en doorzoeken van beslisregels.

3.3 UITPUTTEND ZOEKEN IN BESLISREGELS

3.3.1 Een methode voor het genereren van beslisregels

Zoals bij beslisbomen de complexiteit in het aantal beslisknopen tot uitdrukking wordt gebracht, is bij beslisregels de lengte (d.w.z. het aantal gecombineerde kenmerktests) een maat voor de complexiteit. Een uitputtende benadering bestaat er dan ook uit alle mogelijke beslisregels van een vooraf opgegeven lengte, k, te genereren. De klasse van beslisregels waartoe in hoofdzaak wordt beperkt, bestaat uit die welke wordt voortgebracht door alle mogelijke Booleaanse expressies in DNF-vorm.

Om alle mogelijke beslisregels van lengte k te genereren, worden alle mogelijke expressies van k kenmerken gegenereerd, zonder daarbij logisch equivalente expressies te herhalen. Voor elke gegenereerde expressie worden in een volgende stap de drempelwaarden van de respectieve kenmerken geïnstantieerd. De drempelwaarden worden per kenmerk voorafgaand aan het genereren van de expressies bepaald. De wijze waarop dit plaatsvindt zal later meer in detail worden beschreven.

3.3.2 Complexiteitsanalyse

Om een indicatie van de complexiteit van het uitputtend genereren van beslisregels te krijgen, wordt wederom verondersteld dat een voorbeeldobject door n kenmerkwaarden wordt beschreven en dat beslisregels van lengte k moeten worden gegenereerd. Voorts worden wederom rekenkundige operatoren ($\leq, >$) toegestaan en wordt aangenomen dat elk kenmerk c potentiële drempelwaarden heeft.

Generatie van expressies

Het aantal beslisregels dat ontstaat door alle n^k expressies van k met elkaar door de EN-operator gecombineerde kenmerken—in het vervolg k-conjuncties genoemd—op alle mogelijke manieren te instantiëren bedraagt dan[2]

$$n^k \times 2^k \times c^k. \tag{3.4}$$

Indien DNF-expressies met zowel EN- als OF-operatoren worden beschouwd, dan dient de factor n^k in formule (3.4) met het $k$de Bell-getal, B_k, te worden vermenigvuldigd. Het

[2]Wanneer het aantal potentiële drempelwaarden per kenmerk c_i bedraagt, dan moet de term c^k worden vervangen door het k-voudige produkt van c_i voor alle in de expressie ingevulde kenmerken X_i.

$k$de Bell-getal geeft het aantal manieren weer waarop een verzameling van k elementen in een verzameling van disjuncte deelverzamelingen kan worden gepartitioneerd. Het is recursief gedefinieerd als

$$B_{k+1} = \sum_{i=0}^{k} \binom{k}{i} B_i, \qquad (k \geq 0) \tag{3.5}$$

waarbij B_0 per definitie gelijk is aan 1 [Andrews, 1976]. Voor $k = 0, 1, 2, 3, \ldots$ geldt dat $B_{k+1} = 1, 2, 5, 15, \ldots$

Het totaal aantal beslisregels ligt hiermee in de orde van

$$n^k \times 2^k \times c^k \times B_k. \tag{3.6}$$

Door de commutatieve eigenschap van de EN-operator zijn een aantal van de n^k k-conjuncties logisch gezien echter equivalent. In bijlage A.3 is afgeleid dat, wanneer equivalente expressies buiten beschouwing worden gelaten, er $\binom{n+k-1}{k}$ k-conjuncties in dit opzicht uniek te noemen zijn. Het zelfde argument geldt voor DNF-expressies met zowel EN- als OF-operatoren, zij het dat ook hier een aantal van de $\binom{n+k-1}{k} \times B_k$ expressies logisch gezien equivalent is. Het aantal unieke DNF-expressies is derhalve kleiner, waardoor uiteraard ook het totaal aantal beslisregels dat op grond hiervan wordt voortgebracht (formule (3.6)) navenant kleiner is.

Instantiatie van beslisregels

Om beslisregels te verkrijgen, worden de unieke DNF-expressies in fasen geïnstantieerd; in een eerste fase wordt met elk kenmerk van een expressie één van twee rekenkundige operatoren geassocieerd, waarna in een tweede fase drempelwaarden worden ingevuld. Deze fasen worden achtereenvolgens uitgevoerd, om zodoende systematisch alle mogelijke beslisregels te genereren.

In de eerste fase ontstaan echter combinaties die eerder zijn gegenereerd. Indien, bij wijze van voorbeeld, expressies van twee kenmerken A, B worden beschouwd, dan is de expressie A EN A uniek. Afhankelijk van welke twee rekenkundige operatoren met de respectieve kenmerken worden geassocieerd, levert deze expressie een zinvolle combinatie op. Bij gelijke operatoren ontstaat immers een combinatie die reeds bij het genereren van expressies van één kenmerk is voortgebracht; zoals $(A > ?)$ EN $(A > ?) = (A > ?)$ bijvoorbeeld. Meer in het algemeen geldt dat het aantal zinvolle operatorcombinaties kleiner is dan de factor 2^k in formule (3.6) suggereert. Om na te gaan hoeveel combinaties zinvol zijn, wordt verondersteld dat een k-conjunctie bestaat uit m verschillende kenmerken en dat kenmerk X_i daarin m_i keer voorkomt. Als $m_i = 1$ dan zijn twee operatorcombinaties zinvol, namelijk $(X_i \leq ?)$ en $(X_i > ?)$, en voor $m_i > 1$ slechts drie: $(X_i \leq ?)$, $(X_i > ?)$, en $(X_i \leq ?)$ EN $(X_i > ?)$. Het aantal zinvolle combinaties is derhalve gelijk aan

$$2^p \times 3^q, \tag{3.7}$$

waarin $p = \|\{X_i|m_i = 1\}\|$ en $q = \|\{X_i|m_i > 1\}\|$, het aantal kenmerken dat precies één keer, respectievelijk meer dan één keer voorkomt. Twee interessante gevallen doen zich hierbij voor: voor $p = k$ (d.w.z. elk kenmerk komt precies één keer voor) is $q = 0$ en is formule (3.7) gelijk aan 2^k, terwijl voor $p = 0$ (d.w.z. elk van de m kenmerken komt meer dan één keer voor) $q = \lfloor \frac{k}{m} \rfloor$ en formule (3.7) tot $3^{\lfloor \frac{k}{m} \rfloor}$ reduceert.

Indien eerder gegenereerde combinaties buiten beschouwing worden gelaten en in de tweede fase in de resterende combinaties op alle mogelijke manieren drempelwaarden worden ingevuld, ontstaan echter een aantal beslisregels die semantisch gezien niet zinvol zijn. Veronderstel daartoe dat de c potentiële drempelwaarden λ_i van kenmerk A geordend zijn volgens: $\lambda_1 < \lambda_2 < ... < \lambda_c$. De geïnstantieerde regels

$$(A \leq \lambda_i) \text{ EN } (A > \lambda_j), \qquad i < j \tag{3.8}$$

$$(A \leq \lambda_i) \text{ OF } (A > \lambda_j), \qquad i \geq j \tag{3.9}$$

$$(A \leq \lambda_i) \text{ OF } (A > \lambda_i \text{ EN } A \leq \lambda_j) \tag{3.10}$$

zijn dan bijvoorbeeld niet zinvol, omdat regel (3.8) geen enkel object bedekt, regel (3.9) alle objecten bedekt, en regel (3.10) reduceert tot $(A \leq \lambda_j)$ als $i < j$ en tot $(A \leq \lambda_i)$ als $i \geq j$.

De conclusie van voorgaande uiteenzetting is, dat het geenszins een eenvoudige opgave is om alle verschillende, betekenisvolle, niet-reduceerbare DNF-beslisregels te genereren. Het is overigens een bekend probleem dat het reduceren van een Booleaanse expressie tot zijn meest eenvoudige vorm over het algemeen een niet-triviale zaak is [Flegg, 1965].

In eerste instantie is daarom gekozen voor een rechtstreekse benadering, waarbij eenvoudigweg alle mogelijke combinaties worden voortgebracht. Alhoewel op grond van voorgaande overwegingen duidelijk is dat het totaal aantal zinvolle DNF-beslisregels aanmerkelijk kleiner is dan formule (3.6) suggereert, verschaft een dergelijke benadering inzicht in de wijze waarop de afzonderlijke parameters de omvang van de zoekruimte beïnvloeden. Dit inzicht vormt de basis voor het ontwikkelen van heuristieken die de zoekruimte in omvang doen beperken.

3.3.3 De doelmatigheid van beslisregels

Alvorens nader op de heuristieken in te gaan, is het zinvol de eerder genoemde doelmatigheid van het representatieformalisme van beslisregels te illustreren aan de hand van een vergelijk tussen de complexiteit van het uitputtend genereren van beslisbomen en die van het uitputtend genereren van beslisregels.

Om een eenvoudige expressie als A OF B te verkrijgen, zijn volgens de beslisboombenadering van paragraaf 3.2 bomen met tenminste twee beslisknopen noodzakelijk. Het aantal bomen dat volgens formule (3.3) met $n = k = 2$ wordt voortgebracht om de genoemde expressie te genereren, bedraagt 32. Volgens formule (3.6) daarentegen, worden slechts 8 regels voortgebracht. Dit voorbeeld illustreert dat voor gelijke waarden van

k de factor b_k in formule (3.3) van grotere invloed is dan de factor B_k in formule (3.6). Oorzaak hiervoor is de wederzijdse uitsluiting waarmee de boom behept is. De beslisboom is een 'simultane' classificator voor twee klassen; een object wordt expliciet aan één van beide klassen (K_1, K_2) toegewezen, terwijl de beslisregel een classificator is voor één klasse; indien een object aan de condities daarvan voldoet, krijgt het het corresponderende klasselabel toegewezen, terwijl geen uitspraak wordt gedaan wanneer het niet aan de condities voldoet. Bij een tweeklassenprobleem wordt het object in dat geval doorgaans aan de andere klasse toegekend.

De complexiteit van de beslisboom-benadering neemt nog verder toe ten opzichte van die van beslisregels wanneer de factor k in formule (3.3) groter is dan die in formule (3.6). Dit fenomeen doet zich voor wanneer meer beslisknopen nodig zijn om een expressie in de vorm van een beslisboom te representeren dan in de vorm van een beslisregel. Het voorbeeld van de expressie (A EN B) OF (C EN D) op pagina 31 is hiervoor illustratief. Aan de hand van formule (3.6) en (3.3) kan worden nagegaan dat er 3840 beslisregels worden voortgebracht om de genoemde expressie te genereren, terwijl er ruim 20×10^6 beslisbomen tegenover staan. Het *replication problem* van beslisbomen (zie pagina 31) valt dus duidelijk in het nadeel daarvan uit.

3.3.4 Heuristieken

Eerder (sub 3.3.2) is een ordegrootte-indicatie afgeleid voor de rekenkundige complexiteit van het uitputtend genereren van beslisregels. Formule (3.6) brengt deze complexiteit tot uitdrukking in termen van de probleemparameters n (het aantal beschikbare kenmerken), 2 (het aantal rekenkundige operatoren), c (het aantal potentiële drempelwaarden per kenmerk) en k (de regellengte). In het navolgende worden aan de hand van de complexiteitstermen n^k, 2^k, c^k en B_k van formule (3.6) heuristieken geïntroduceerd die ingrijpen op de afzonderlijke termen, met als doel de totale complexiteit te reduceren.

Selectie van kenmerken

Eén van de belangrijkste parameters is het aantal kenmerken dat wordt gebruikt om beslisregels te genereren. Sommige kenmerken hebben een sterker scheidend vermogen dan andere kenmerken en zullen derhalve een grotere bijdrage aan de prestaties van de classificator leveren. Helaas echter is *a priori* vaak niet bekend om welke kenmerken het gaat.

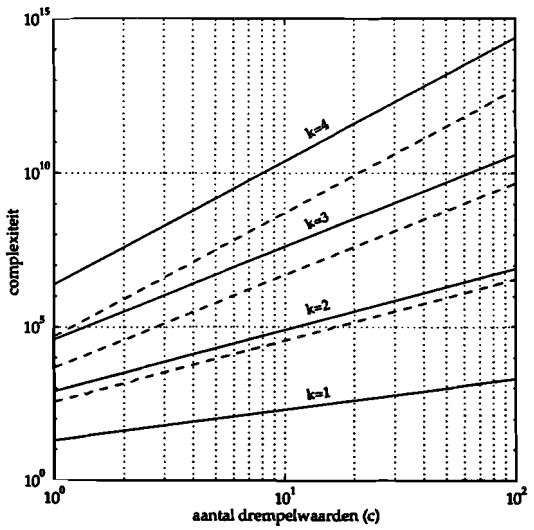
Een eerste mogelijkheid zou kunnen zijn gebruik te maken van expertkennis. Vaak zijn experts in staat om redelijk goed aan te geven welke kenmerken relevant zijn. Echter, indien expertkennis schaars is of het wenselijk is een classificator zonder inmenging van een expert te construeren, bijvoorbeeld omdat een expert geneigd is aan bepaalde kenmerken een zekere 'voorkeur' te geven, is een meer formele benadering nodig.

Het probleem om de beste deelverzameling van kenmerken te vinden is een bekend en moeilijk statistisch probleem [Duda & Hart, 1973, pagina's 246–248]. Voor sommige statistische problemen echter, kunnen *branch and bound* technieken een optimale deelverzameling van k kenmerken bepalen [Narenda & Fukunaga, 1977]. Deze technieken beginnen met alle n beschikbare kenmerken, elimineren één kenmerk per stap, en evalueren de prestaties van de classificator voor een deelverzameling van k kenmerken. Zij zijn echter alleen optimaal in termen van bepaalde statistische afstandsmaten en niet in termen van de eerdergenoemde prestatiematen. Voor beslisregels zijn dergelijke technieken derhalve niet geschikt.

Voortbouwend op het voorgaande, is het een mogelijkheid om uit een verzameling van n beschikbare kenmerken alle mogelijke deelverzamelingen van k kenmerken te beschouwen. Deze eerste heuristiek vervangt de term n^k door $\binom{n}{k}$, waardoor de complexiteitsformule de volgende gedaante krijgt:

$$\binom{n}{k} \times 2^k \times c^k \times B_k. \tag{3.11}$$

Vooral voor grotere regellengten geeft dit een enorme reductie van het aantal mogelijkheden. In figuur 3.5 is deze reductie voor $n = 10$ en $k = 1, 2, 3$ en 4 grafisch weergegeven. Langs de horizontale as is het aantal potentiële drempelwaarden c, en langs de verticale



Figuur 3.5 Complexiteitsreductie door heuristiek.

as de complexiteit uitgezet. De ononderbroken en onderbroken lijnen hebben respectievelijk betrekking op het aantal mogelijkheden zonder, en met de heuristiek.

Toepassing van deze heuristiek komt er op neer dat elk kenmerk ten hoogste één keer per beslisregel voorkomt. Voor kenmerken met een unimodale verdeling hoeft

deze beperking niet bezwaarlijk te zijn, omdat de populaties met één beslisdrempel goed gescheiden kunnen worden. Voor multimodale verdelingen daarentegen, kan de beperking die deze heuristiek oplegt echter wel bezwaarlijk zijn, vanwege de behoefte aan meerdere beslisdrempels per kenmerk. Indien het daarom wenselijk is om regels te genereren waarbij een kenmerk meer dan eens per regel kan voorkomen, dan is deze heuristiek per definitie niet van toepassing, doch kan formule (3.11) als ondergrens van het aantal mogelijke beslisregels worden aangemerkt.

In hoofdstuk 4 zal overigens blijken dat deze heuristiek in de praktijk nauwelijks afbreuk doet aan de optimaliteit van de oplossing.

Selectie van rekenkundige operatoren

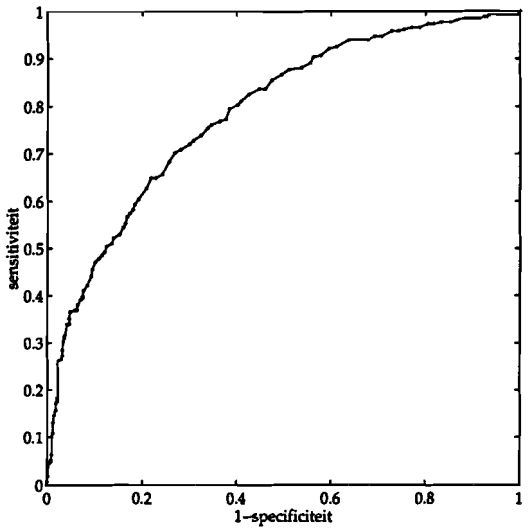
Een volgende parameter is het aantal rekenkundige operatoren dat in een kenmerktest wordt toegestaan. Alhoewel deze van minder grote betekenis is voor de totale complexiteit, introduceert zij een exponentiële term in de complexiteitsformule. Het is derhalve voordelig wanneer deze term gereduceerd of geëlimineerd kan worden. Eliminatie is mogelijk door bij elk kenmerk één bepaalde operator te associëren. In dat geval wordt de complexiteitsterm 2^k door 1^k vervangen en krijgt de complexiteitsformule de gedaante

$$n^k \times c^k \times B_k. \tag{3.12}$$

Er zijn een aantal mogelijkheden om per kenmerk één rekenkundige operator te bepalen.

Een eerste mogelijkheid bestaat er uit voor beide klassen de mediaan van de kenmerkwaarden te bepalen en de $\leq$ operator te kiezen wanneer de mediaan van de klasse waarvoor een beslisregel wordt geïnduceerd de kleinste van de twee is, en anders de $>$ operator te kiezen [Weiss *et al.*, 1987].

Een tweede mogelijkheid is gebruik te maken van ROC-curven [Green & Swets, 1966; Beck & Shultz, 1986]. Een ROC-curve wordt geconstrueerd door voor alle mogelijke drempelwaarden van een dichotome kenmerktest de sensitiviteit als functie van de fout-positief fractie (1-specificiteit) weer te geven. Een voorbeeld hiervan is in figuur 3.6 op pagina 38 afgebeeld. Bij een perfecte test is het aantal fout-positieve en fout-negatieve beslissingen nul, waardoor de ROC-curve bestaat uit een lijnstuk van het punt $(0,0)$ naar $(0,1)$ en van $(0,1)$ naar $(1,1)$. Een test waarbij de fractie fout-positieve en correct-positieve beslissingen gelijke tred houdt, wordt afgebeeld als een diagonaal lijnstuk van het punt $(0,0)$ naar $(1,1)$. Derhalve zal een zinvolle test een ROC-curve hebben die merendeels in het boven-driehoeksgebied $(0,0)$, $(1,1)$, $(0,1)$ ligt. Wanneer de ROC-curve van een kenmerktest geheel boven of onder de diagonaal ligt, dan ligt daarmee de rekenkundige operator vast. Immers, als de ROC-curve geheel boven de diagonaal ligt, dan is voor elke drempelwaarde de fractie correct-positieve beslissingen groter dan de fractie fout-positieve beslissingen en kan worden nagegaan dat, wanneer de curve voor toenemende drempelwaarden van het punt $(0,0)$ in de richting van het punt $(1,1)$ wordt doorlopen, door de test een dichotomie is aangebracht waarbij het testresultaat positief (negatief)



Figuur 3.6 Voorbeeld van een empirische ROC-curve.

wordt genoemd voor kenmerkwaarden kleiner (groter) dan de drempelwaarde. In geval voor een positief (negatief) testresultaat een beslisregel wordt afgeleid, dient derhalve de $\leq$ ($>$) operator te worden gekozen. Wordt de ROC-curve daarentegen voor toenemende drempelwaarden in de tegenovergestelde richting doorlopen, dan is door de test een dichotomie aangebracht waarbij het testresultaat positief (negatief) wordt genoemd voor kenmerkwaarden groter (kleiner) dan de drempelwaarde. Derhalve dient in dit geval de $>$ ($\leq$) operator te worden gekozen om een beslisregel voor een positief (negatief) testresultaat af te leiden.

In [Bamber, 1975] is aangetoond dat het oppervlak W onder een ROC-curve een niet-parametrische schatting is voor de waarschijnlijkheid θ dat, gegeven één willekeurig object van de klasse 'normaal' met kenmerkwaarde x_N en één willekeurig object van de klasse 'abnormaal' met kenmerkwaarde x_A, de kenmerkwaarde van het abnormale object groter is dan van het normale object:

$$W = \Pr\{x_A > x_N\}. \tag{3.13}$$

Omdat geen aannamen over de vorm van de verdelingen van de objecten worden gemaakt, kan het oppervlak W onder de curve worden geïnterpreteerd als een maat voor de waarschijnlijkheid van een correcte rangschikking (d.w.z. $x_A > x_N$ of $x_A < x_N$) van een normaal/abnormaal-paar. Zo staat voor $W = 1$ bijvoorbeeld met zekerheid vast hoe de objecten georiënteerd zijn: alle normale objecten liggen langs de kenmerkas rechts van alle abnormale objecten. Voor $W = \frac{1}{2}$ wisselen de objecten van de verschillende klassen elkaar af. Zodoende kan op grond van de waarde van W een rekenkundige operator worden gekozen die past bij de oriëntatie van de twee populaties. Soms is het

echter wenselijk om voor een kenmerk toch twee rekenkundige operatoren toe te staan, bijvoorbeeld wanneer de verdeling van kenmerkwaarden bimodaal is of $W > \frac{1}{2}$ maar de ROC-curve niet geheel boven of onder de diagonaal ligt. In dat geval wordt waar mogelijk één operator gekozen en worden anders beide operatoren toegestaan.

In hoofdstuk 4 wordt een experiment beschreven waarin empirisch wordt nagegaan in hoeverre deze heuristiek afbreuk doet aan de optimaliteit van de oplossing en worden de resultaten van de verschillende methoden met elkaar vergeleken.

Reductie van het aantal drempelwaarden

De meest rekenintensieve term van de complexiteitsformule is doorgaans c^k, omdat het aantal potentiële drempelwaarden vaak vele malen groter is dan het aantal beschikbare kenmerken en rekenkundige operatoren. Het aantal potentiële drempelwaarden is gelukkig veelal aanmerkelijk kleiner dan het totaal aantal beschikbare voorbeeldobjecten, omdat kan worden aangetoond dat, wanneer de objecten op kenmerkwaarden worden geordend, interessante drempelwaarden zich uitsluitend voordoen tussen twee opeenvolgende objecten met een verschillend klasselabel [Fayyad & Irani, 1992]. Desalniettemin is het voordelig wanneer het aantal potentiële drempelwaarden nog verder kan worden gereduceerd.

Een heuristiek om dit te bewerkstelligen is een minimum te stellen aan het aantal objecten tussen twee opeenvolgende beslisdrempels. Zo wordt voorkomen dat onnodig veel beslisdrempels ontstaan in gebieden waar univariaat gezien de objecten van verschillende klassen elkaar frequent afwisselen.

In hoofdstuk 4 wordt een experiment beschreven waarbij wordt nagegaan welk effect deze heuristiek heeft op de complexiteit en hoe de prestaties er door worden beïnvloed.

De vorm van beslisregels

De term B_k tenslotte, kan uit de complexiteitsformule worden geëlimineerd wanneer enkel conjunctieve regels dienen te worden gegenereerd in plaats van DNF-regels met zowel EN- als OF-operatoren. Het argument om deze heuristiek daadwerkelijk te gebruiken veronderstelt echter de beschikbaarheid van achtergrondkennis. Vaak is deze kennis niet voorhanden of is het niet wenselijk haar aan te wenden.

Voorts is het mogelijk om regels niet alleen in DNF-vorm, maar ook in CNF-vorm te genereren. Het is eenvoudig na te gaan dat het uitputtend genereren van beslisregels in CNF-vorm een probleem is van dezelfde complexiteitsorde als dat van het uitputtend genereren van regels in DNF-vorm. Wanneer het wenselijk is om regels in beide vormen te genereren, dan kan de totale complexiteit worden gereduceerd door dubbelen buiten beschouwing te laten.

Combinatie van heuristieken

Vanzelfsprekend is het mogelijk om meerdere heuristieken tegelijkertijd toe te passen. De respectieve complexiteitstermen van formule (3.6) dienen dan door de hierboven aangegeven termen te worden vervangen. Wanneer in een experiment bijvoorbeeld alle DNF-regels van lengte k of kleiner moeten worden gegenereerd, elk kenmerk ten hoogste één maal voorkomt, een rekenkundige operator op grond van de mediaan-heuristiek wordt bepaald en het aantal drempelwaarden tot c_* is gereduceerd, dan is de complexiteit van de orde

$$\sum_{i=1}^{k} \binom{n}{i} \times c_*^i \times B_i. \tag{3.14}$$

3.3.5 Het bepalen van de optimale regellengte

Omdat de hiervoor beschreven methode zoekt naar de enkele 'beste' regel van een vooraf opgegeven lengte, kan zij bij uitstek met *resampling*technieken worden gecombineerd. Resampling beperkt zich immers niet tot het schatten van prestatiematen alleen, maar kan worden gebruikt om elke willekeurige parameter van een populatie te schatten [Efron & Gong, 1983].

Omdat de zoekmethode niet direct aangeeft voor welke regellengte de beste prestaties worden behaald, kan zij in combinatie met resamplingtechnieken worden gebruikt om een schatting te bepalen van de regellengte waarbij de te maximaliseren prestatiemaat zijn grootste waarde aanneemt. Alhoewel bij toenemende regellengte de resubstitutieschatting een steeds optimistischer resultaat geeft, bestaat het gevaar voor overspecialisatie, waardoor de prestaties op een onafhankelijk testbestand vanaf een zekere regellengte zullen afnemen. Resamplingtechnieken kunnen worden gebruikt om betrouwbare schattingen te bepalen van de prestaties van een regel met een specifieke regellengte.

Zo omvat een op *cross-validation* gebaseerde werkwijze voor het bepalen van de optimale regellengte de stappen van:

1. het voor elke regel van lengte l bepalen van een geschatte waarde CV_l van de te optimaliseren prestatiemaat;

2. het kiezen van de optimale regellengte k waarvoor CV_k het minimum is; en

3. het voor de optimale regellengte k bepalen van de beste regel op basis van alle n beschikbare voorbeeldobjecten.

Sub 2.4.2 is aangegeven dat ook het PVM-algoritme deze werkwijze gebruikt om de optimale regellengte te bepalen.

3.4 CONCLUSIES

De in de inleiding genoemde bezwaren van conventionele inductietechnieken kunnen grotendeels worden ondervangen door kenmerktests niet alleen uitvoeriger te combineren, maar vooral ook door ze voor meerdere drempelwaarden te evalueren. Een eerste benadering bestaat derhalve uit het uitputtend genereren van alle mogelijke beslisboom- of beslisregelclassificatoren. Uit de analyse van de rekenkundige complexiteit blijkt dat het genereren van beslisregels efficiënter is. Desalniettemin zal het in de praktijk vaak nodig zijn om, als gevolg van de combinatorische explosie die optreedt, heuristieken te gebruiken om de omvang van de zoekruimte tot aanvaardbare proporties te reduceren. Een aantal heuristieken is aan de hand van de afzonderlijke termen van de complexiteitsformule geïntroduceerd, waarbij is aangegeven op welke wijze zij daarop ingrijpen.

In het volgende hoofdstuk worden een aantal experimenten beschreven met gegevensbestanden die eerder in de literatuur geanalyseerd zijn. Het doel daarvan is de invloed van de heuristieken op de complexiteit en de prestaties van de resulterende classificator na te gaan.

HOOFDSTUK 4

Experimenten

4.1 INLEIDING

In het vorige hoofdstuk is een nieuwe inductiemethode beschreven en geanalyseerd, waarmee het mogelijk is een classificator te construeren die optimaal is onder voorwaarden die aan de diagnostische prestatiematen van de resulterende classificator zijn opgelegd. In dit hoofdstuk wordt, aan de hand van een aantal uit de literatuur bekende gegevensbestanden, experimenteel onderzocht in hoeverre de geïntroduceerde heuristieken afbreuk doen aan de optimaliteit van de oplossing en hoe de nieuwe methode zich verhoudt tot andere, meer conventionele inductietechnieken.

4.2 SELECTIE VAN GEGEVENSBESTANDEN

De gegevensbestanden waarvan in de experimenten gebruik is gemaakt, zijn op grond van de volgende overwegingen geselecteerd uit een bibliotheek van veelgebruikte bestanden [Murphy & Aha, 1994]:

1. Een bestand heeft bij voorkeur betrekking op een tweeklassenprobleem, omdat de nieuwe methode in eerste instantie daarvoor is ontwikkeld en een strategie voor toepassing op meerklassenproblemen voorshands buiten beschouwing is gelaten.
2. De kenmerkwaarden zijn continu en/of binair, doch niet nominaal, aangezien dit bij de huidige implementatie van het algoritme problemen geeft bij het bepalen van geschikte drempelwaarden.

3. Het aantal kenmerken ligt bij voorkeur in de orde van 10, daar een groter aantal een enorme complexiteitstoename teweegbrengt en geen essentiële bijdrage levert aan de beantwoording van de vraag welk effect de heuristieken op de oplossing hebben.

4. De bestanden hebben een middelgrote omvang en bevatten enkele honderden voorbeeldobjecten, waardoor voor het schatten van de werkelijke prestaties van een classificator *cross-validation* wordt geprefereerd boven *leave-one-out*.

5. Bij voorkeur zijn geen *missing values* aanwezig; anderszins worden de desbetreffende objecten uit het bestand verwijderd.

6. De bestanden zijn bij voorkeur afkomstig uit een medisch domein, zodat met name aandacht kan worden besteed aan de diagnostische prestatiematen en de interpretatie daarvan.

7. Waar mogelijk zijn eerdere classificatieresultaten bekend uit de literatuur.

De hiervoor genoemde overwegingen hebben geleid tot selectie van de volgende gegevensbestanden: BU (Bupa), GL2 (Glass2), HE (Hepatitis), IR (Iris) en PI (Pima Indians Diabetes). Voor een specificatie van deze bestanden wordt verwezen naar bijlage B. De belangrijkste karakteristieken zijn in tabel 4.1 beknopt weergegeven.

Tabel 4.1 Karakteristieken van geselecteerde gegevensbestanden.

Naam	Klassen	Kenmerken	Kenmerkwaarden	Omvang	MVs*
BU	2	6	continu	345	–
GL2	2	9	continu	214	–
HE	2	19	binair/continu	155	75
IR	3	4	continu	150	–
PI	2	8	continu	768	–

**Missing values.*

Tabel 4.1 laat zien dat IR als enige een meerklassenprobleem is [Fisher, 1936]. De reden om dit bestand desondanks te selecteren is dat het betrekking heeft op een relatief eenvoudig discriminatieprobleem, derhalve veelvuldig in de literatuur wordt aangehaald, en classificatietechnieken doorgaans niet worden geaccepteerd wanneer zij slecht op dit bestand presteren.

4.3 OPTIMALITEIT VAN GEÏNDUCEERDE BESLISREGELS

Het is de verwachting dat heuristieken afbreuk zullen doen aan de optimaliteit van de resulterende classificator. Om enig inzicht te verschaffen in de mate waarin dit fenomeen zich voordoet, wordt in deze paragraaf aan de hand van een aantal experimenten onderzocht hoe de heuristieken ingrijpen op het classificatieprobleem, welke gevolgen dit heeft voor de uiteindelijke oplossing, en welke heuristieken 'veelbelovend' blijken te zijn.

4.3.1 Uitputtend zoeken naar optimale regels

Omdat IR een relatief eenvoudig discriminatieprobleem betreft, is het zeer geschikt om te worden gebruikt in experimenten waarbij uitputtend wordt gezocht in de oplossingsruimte van alle mogelijke beslisregels van een vooraf opgegeven lengte. Omdat het hier een drieklassenprobleem betreft, wordt in het eerste experiment de hiernavolgende strategie gevolgd.

Werkwijze voor experiment #1

In drie tweeklassenproblemen worden de klassen 'Iris Setosa', 'Iris Versicolor', en 'Iris Virginica' tegen hun respectieve negaties afgezet en worden in een eerste stap van het experiment regels van toenemende lengte geïnduceerd op basis van het gehele bestand. Alhoewel de prestaties van de resulterende regels hierdoor optimistisch vertekend zijn, zal aan de hand daarvan een aantal interessante aspecten van de inductiemethode worden geïllustreerd.

Door middel van de sub 3.3.5 beschreven werkwijze wordt in een tweede stap van het experiment voor elk van de drie problemen de optimale regellengte bepaald, in aansluiting waarop in een derde stap op basis van het gehele bestand voor die regellengten de optimale regels worden afgeleid. De twee best presterende regels worden gebruikt, terwijl de resterende klasse wordt gekozen wanneer aan geen van beide regels is beantwoord. Er wordt op nauwkeurigheid geoptimaliseerd, waarbij aan de overige vier diagnostische prestatiematen geen restricties zijn opgelegd.

Resultaten van experiment #1

Als resultaat van de eerste stap zijn in tabel 4.2 voor de klasse 'Iris Versicolor' de vier meest veelbelovende regels van lengte 1, 2 en 3 weergegeven, waarbij in de laatste kolom de resubstitutienauwkeurigheid is vermeld.

Geheel volgens verwachting laat tabel 4.2 zien dat de (resubstitutie)prestaties voor toenemende regellengte verbeteren. Daarnaast blijken er kenmerken te zijn waarvoor 'veelbelovende' tests met zowel de $\leq$ als $>$ operator worden voortgebracht; vergelijk (*Petal Length* ≤ 4.85) EN (*Petal Width* > 0.80) met (*Petal Length* > 2.45) EN (*Petal Width* ≤ 1.65). Dit lijkt erop te duiden dat een heuristiek die *a priori* één rekenkundige operator

Tabel 4.2 Resultaten van experiment #1—Veelbelovende regels voor 'Iris Versicolor'.

Regel	Nauwkeurigheid
(*Sepal Width* ≤ 2.55)	0.71333
(*Sepal Width* ≤ 2.65)	0.72000
(*Sepal Width* ≤ 2.75)	0.72667
(*Sepal Width* ≤ 2.95)	0.74000
(*Petal Length* ≤ 4.75) EN (*Petal Width* > 0.80)	0.95333
(*Petal Length* ≤ 4.85) EN (*Petal Width* > 0.80)	0.95333
(*Petal Length* > 2.45) EN (*Petal Width* ≤ 1.65)	0.96000
(*Petal Length* > 2.45) EN (*Petal Width* ≤ 1.75)	0.96000
(*Sepal Length* ≤ 6.95) EN (*Petal Length* > 2.45) EN (*Petal Width* ≤ 1.65)	0.96000
(*Sepal Length* ≤ 6.95) EN (*Petal Length* > 2.45) EN (*Petal Width* ≤ 1.75)	0.96000
(*Sepal Length* ≤ 7.05) EN (*Petal Length* > 2.45) EN (*Petal Width* ≤ 1.65)	0.96667
(*Sepal Length* ≤ 7.05) EN (*Petal Length* > 2.45) EN (*Petal Width* ≤ 1.75)	0.96667

bepaalt weinigbelovend zou kunnen zijn. Zo gaat het in geval van de twee hiervoor als voorbeeld genoemde conjuncties mis wanneer één van beide operatoren anders wordt gekozen. In een volgend experiment zal de invloed van deze heuristiek daarom nader worden geanalyseerd.

Voor het IR-probleem zijn de meest veelbelovende beslisregels van lengte 1, 2 en 3 per klasse in tabel 4.3 op pagina 47 weergegeven. De laatste kolom vermeldt wederom de resubstitutienauwkeurigheid.

Uit de tabellen 4.2 en 4.3 volgt een tweetal interessante constateringen. Niet alleen blijkt een kenmerk ten hoogste één maal per beslisregel voor te komen, maar ook doet het fenomeen zich voor dat veelbelovende langere regels vaak door combinatie van kortere regels worden verkregen.

De in de tweede stap van het experiment door *cross-validation* bepaalde schattingen van de nauwkeurigheid zijn in figuur 4.1 op pagina 48 als functie van de regellengte grafisch weergegeven.

De klasse 'Iris Setosa' blijkt uit figuur 4.1 met een enkele kenmerktest perfect van de andere twee klassen te kunnen worden gescheiden. Voor de klasse 'Iris Versicolor' blijkt een regellengte 2, en voor 'Iris Virginica' een regellengte 3 optimaal te zijn.

De optimale regels die uit de derde stap op grond van het gehele bestand zijn geconstrueerd, zijn voor de optimale regellengte uiteraard gelijk aan die welke in de eerste stap zijn gevonden. Omdat de *cross-validation* nauwkeurigheid van de optimale regel voor de klasse 'Iris Virginica' (0.96190) groter is dan die voor 'Iris Versicolor' (0.94047), kunnen

Tabel 4.3 Resultaten van experiment #1—Veelbelovende DNF-regels voor het IR-probleem.

Regel	Nauwkeurigheid
'Iris Setosa'	
(*Petal Length* ≤ 2.45)	1.00000
(*Petal Width* ≤ 0.80)	1.00000
'Iris Versicolor'	
(*Sepal Width* ≤ 2.95)	0.74000
(*Petal Length* > 2.45) EN (*Petal Width* ≤ 1.65)	0.96000
(*Petal Length* > 2.45) EN (*Petal Width* ≤ 1.75)	0.96000
(*Sepal Length* ≤ 7.05) EN (*Petal Length* > 2.45) EN (*Petal Width* ≤ 1.65)	0.96667
(*Sepal Length* ≤ 7.05) EN (*Petal Length* > 2.45) EN (*Petal Width* ≤ 1.75)	0.96667
'Iris Virginica'	
(*Petal Width* > 1.65)	0.96000
(*Petal Width* > 1.75)	0.96000
(*Petal Length* > 4.95) OF (*Petal Width* > 1.65)	0.98000
(*Sepal Width* ≤ 3.05) EN (*Petal Length* > 4.95) OF (*Petal Width* > 1.55)	0.98667
(*Sepal Width* ≤ 3.05) EN (*Petal Length* > 4.95) OF (*Petal Width* > 1.65)	0.98667
(*Sepal Width* ≤ 3.15) EN (*Petal Length* > 4.95) OF (*Petal Width* > 1.55)	0.98667
(*Sepal Width* ≤ 3.15) EN (*Petal Length* > 4.95) OF (*Petal Width* > 1.65)	0.98667

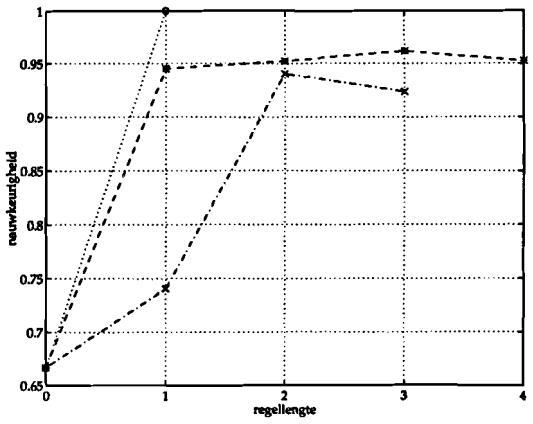
voor classificatie de volgende twee optimale beslisregels worden gebruikt:

als (*Petal Length* ≤ 2.45) **dan** 'Iris Setosa', (4.1)

als (*Sepal Width* ≤ 3.15) EN (*Petal Length* > 4.95) OF (*Petal Width* > 1.65)
dan 'Iris Virginica'. (4.2)

De uiteindelijke nauwkeurigheid van de classificator voor IR is daarmee gelijk aan (0.96190). Uit tabel B.1 op pagina 84 volgt dat dit resultaat superieur is aan het voor het PVM-algoritme in de literatuur gepubliceerde resultaat. In [Weiss *et al.*, 1987] is een experiment beschreven waarin uitputtend is gezocht naar optimale regels van lengte 2 of kleiner. Naast regel (4.1) wordt als alternatief voor regel (4.2) de regel

als (*Petal Length* > 5) OF (*Petal Width* > 1.7) **dan** 'Iris Virginica'



Figuur 4.1 Nauwkeurigheid van beslisregels voor klasse 'Iris Setosa' (o), 'Iris Versicolor' (×) en 'Iris Virginica' (*).

gevonden met een *cross-validation* nauwkeurigheid van (0.96000). Eerder was niet bekend dat hiervoor een langere, beter presterende regel blijkt te bestaan [Holte, 1993, pagina 88].

Hiermee is voor IR aangetoond dat met het EXPLORE-algoritme minstens zo goede beslisregels als uit de literatuur bekend kunnen worden geïnduceerd. Voorts wordt geconcludeerd dat, wanneer een goede oplossing in de vorm van een relatief korte regel bestaat, de nieuwe methode een mogelijkheid biedt om nieuwe, concurrerende beslisregels af te leiden.

4.3.2 Uitputtend versus heuristisch zoeken naar optimale regels

Om na te gaan welk effect de heuristieken op de in de vorige subparagraaf aangegeven oplossing hebben, zijn de volgende twee experimenten uitgevoerd. In experiment #2 wordt aan de hand van IR de invloed geanalyseerd van een eerste heuristiek waarmee een kenmerk ten hoogste één maal per beslisregel kan voorkomen, en van een tweede heuristiek die *a priori* bij elk kenmerk één rekenkundige operator bepaalt. Het effect van een derde heuristiek, die het aantal drempelwaarden reduceert, laat zich aan de hand van dit bestand niet zo duidelijk illustreren, omdat het aantal potentiële drempelwaarden per kenmerk relatief gering is. Aan de hand van twee andere gegevensbestanden wordt in experiment #3 de invloed van deze heuristiek op de complexiteit en de prestaties van de resulterende classificator nader geanalyseerd.

Werkwijze voor experiment #2

In een eerste stap van het experiment wordt de eerste heuristiek gebruikt. Op grond van de sub 4.3.1 afgeleide regels, waaruit blijkt dat een kenmerk niet meer dan eens per regel

voorkomt, volgt dat deze heuristiek geen afbreuk aan de oplossing doet. Deze heuristiek wordt derhalve zonder bezwaar gebruikt in een tweede stap van het experiment, waarin de tweede heuristiek voor drie methoden wordt geëvalueerd: de eerste methode is gebaseerd op het verschil van medianen van de kenmerkwaarden, de tweede op het oppervlak W onder de ROC-curve, en de derde op het boven of onder de diagonaal liggen van de ROC-curve (zie pagina 37 en 39).

Resultaten van experiment #2

In tabel 4.4 is de nauwkeurigheid voor regels van verschillende lengte zowel voor uitputtend als heuristisch zoeken weergegeven. De eerste, tweede en derde methode waarop de tweede heuristiek is gebaseerd, worden respectievelijk aangeduid als Mediaan-, W- en Diagonaal-methode.

Tabel 4.4 Resultaten van experiment #2—*Cross-validation* nauwkeurigheid.

	Uitputtend	Heuristisch		
		Mediaan	W	Diagonaal
'Iris Setosa'				
1	1.00000	1.00000	1.00000	1.00000
'Iris Versicolor'				
1	0.74000	0.74000*	0.74000	0.74000
2	0.94047*	0.71333	0.75358*	0.94047*
3	0.92381	–	0.73571	0.92381
'Iris Virginica'				
1	0.94523	0.94523	0.94523	0.94523
2	0.95238	0.96000*	0.95238	0.95238
3	0.96190*	0.96000	0.96190*	0.96190*
4	0.95238	0.94523	0.95238	0.95238

*Nauwkeurigheid voor de optimale regellengte.

Uit tabel 4.4 blijkt dat het kiezen van één rekenkundige operator afbreuk aan de optimaliteit van de oplossing kan doen. Zo blijkt voor de klasse 'Iris Versicolor' dat zowel bij het uitputtend als heuristisch zoeken naar regels van lengte 1 de kenmerktest (*Sepal Width* ≤ 2.95) consequent als beste wordt aangemerkt. Voor regellengte 2 echter, blijkt de heuristiek op basis van de Mediaan- en W-methode de prestaties nadelig te beïnvloeden. Klaarblijkelijk is door de heuristiek bij één van de twee kenmerken uit de

als meest veelbelovende aangemerkte regels van lengte 2 (zie tabel 4.3) de 'verkeerde' operator gekozen. Nader onderzoek bevestigt dat volgens de Mediaan-methode bij het kenmerk *'Petal Length'* de ≤ operator is gekozen, en dat volgens de *W*-methode bij het kenmerk *'Petal Width'* de > operator is gekozen. Dit verklaart tevens de onderlinge verschillende prestaties. De laatste kolom van de tabel geeft aan dat de Diagonaal-methode uitkomst kan bieden; de resultaten van de heuristische benadering komen overeen met die van de uitputtende benadering, terwijl de complexiteit gereduceerd is door waar mogelijk één rekenkundige operator te kiezen. Overigens wordt opgemerkt dat geen van deze heuristieken effect heeft op de uiteindelijke twee als beste geselecteerde beslisregels.

Werkwijze voor experiment #3

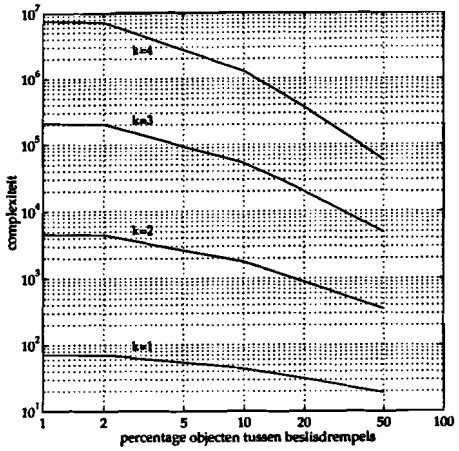
Het BU-bestand en HE-bestand bevatten kenmerken met relatief veel (d.w.z. enkele tientallen) potentiële drempelwaarden per kenmerk. In een serie experimenten waarbij een deelverzameling van beslisdrempels wordt bepaald door voor verschillende waarden een minimum te stellen aan het aantal objecten dat tussen twee opeenvolgende beslisdrempels ligt, wordt de complexiteit(sreductie) afgezet tegen (de afname van) de nauwkeurigheid. Daartoe wordt voor verschillende regellengte zowel de complexiteit als de resubstitutienauwkeurigheid grafisch uitgezet als functie van het minimum aantal objecten tussen twee opeenvolgende beslisdrempels. Aangezien absolute aantallen in dit verband weinig informatief zijn en de vergelijkbaarheid bemoeilijken, worden de aantallen gerelateerd aan het totaal aantal objecten in het gegevensbestand. Door deze maatregel is het aantal beslisdrempels dat resteert, ook voor bestanden van verschillende omvang van vergelijkbare ordegrootte. De complexiteit en nauwkeurigheid zijn beide bepaald voor regels van lengte $k = 1, 2, 3$ en 4 bij percentages van 1, 2, 5, 10, 20 en 50%.

Resultaten van experiment #3

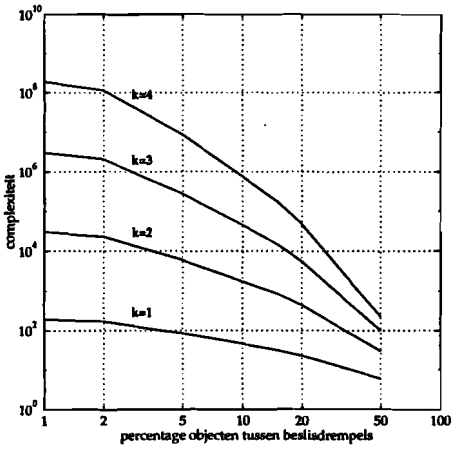
De figuren 4.2 en 4.3 op pagina 51 laten voor de verschillende regellengten zien dat de complexiteit sterk afneemt wanneer volgens de derde heuristiek een toenemend aantal minimum objecten tussen opeenvolgende beslisdrempels is vereist.

Het effect dat deze heuristiek op de classificatienauwkeurigheid heeft, is in de figuren 4.4 en 4.5 voor dezelfde regellengten weergegeven.

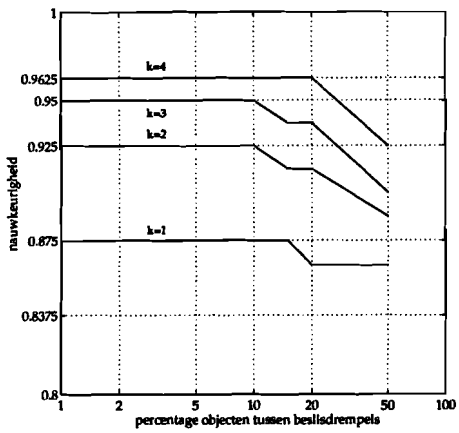
Voor beide bestanden blijkt de nauwkeurigheid nauwelijks af te nemen wanneer het minimum vereiste aantal objecten kleiner is dan circa 5–10% van de totaalomvang van het bestand. Voor grotere percentages echter, weegt de complexiteitsreductie veelal niet op tegen de afname van de prestaties.



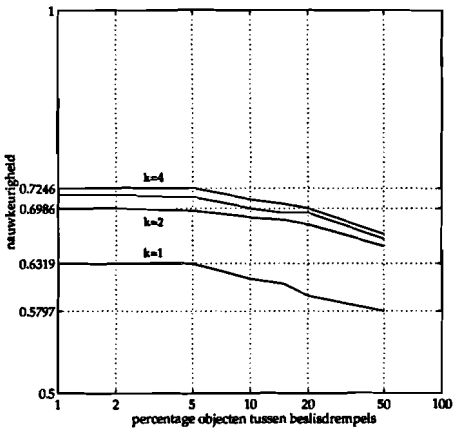
Figuur 4.2 Complexiteit voor HE.



Figuur 4.3 Complexiteit voor BU.



Figuur 4.4 Nauwkeurigheid voor HE.



Figuur 4.5 Nauwkeurigheid voor BU.

Nieuwe heuristieken

Op grond van deze constatering wordt een meerstapsanalyse voor het bepalen van de drempelwaarden voorgesteld. Volgens deze methode kan per kenmerk in een eerste stap het aantal drempelwaarden worden gereduceerd door met behulp van de derde heuristiek het maximum aantal beslisdrempels te specificeren en op grond daarvan alle mogelijke beslisregels te genereren. Veelbelovende beslisregels geven vervolgens een indicatie van 'interessante' intervallen, waarop in een tweede stap kan worden ingezoomd door tussen de twee beslisdrempels die een interessant interval definiëren, een fijnere verdeling aan te brengen. Hierna worden op basis van de zojuist verkregen verzameling drempelwaarden opnieuw alle mogelijke regels gegenereerd. In een derde en volgende stap kan deze tweede stap recursief worden herhaald. Zo ontstaat in eerste instantie op grove schaal een indruk van de oplossingsruimte, waarna in tweede instantie in interessante delen daarvan op fijnere schaal naar betere alternatieven kan worden gezocht.

Een ander voorstel om de rekenkundige complexiteit te reduceren bestaat er uit langere regels te genereren door kortere regels te combineren. De rationale hiervoor is dat, zoals eerder voor IR is gebleken (zie experiment #1) en nu ook voor BU en HE blijkt dat een kenmerk niet meer dan eens per regel voorkomt[1], veelbelovende langere regels vaak worden verkregen door combinatie van veelbelovende kortere regels. Daarnaast zijn de rekenkundige kosten om langere regels te genereren nog altijd zeer hoog. Zo ligt de complexiteit voor het uitputtend genereren van alle mogelijke beslisregels van lengte 2 voor HE bijvoorbeeld in de orde van 5×10^3, hetgeen op een HP 715/80 werkstation circa een halve seconde rekentijd vergt, terwijl de complexiteit voor regellengte 4 in de orde van 7×10^6 ligt en circa 10 minuten vergt. Door een tabel van 'veelbelovende' regels bij te houden, zouden regels van lengte 4 aanzienlijk sneller kunnen worden gegenereerd door bijvoorbeeld regels van lengte 2 met elkaar te combineren. Een dergelijke heuristische benadering is reeds eerder door Weiss *et al.* (1987) voorgesteld en waardevol gebleken.

In de huidige implementatie van het algoritme is voorshands geen rekening gehouden met deze mogelijkheden om de snelheid van het inductieproces te vergroten.

4.4 VERGELIJKING MET CONVENTIONELE INDUCTIETECHNIEKEN

In [Breiman *et al.*, 1984] is aangetoond dat beslisbomen gegenereerd door het CART-algoritme in vergelijking met concurrerende statistische classificatoren zeer goed presteren. Omdat beslisregels nauw verwant zijn aan beslisbomen, mag worden verwacht dat een oplossing in de vorm van beslisregels vergelijkbare resultaten oplevert.

In deze paragraaf wordt voor een aantal van de geselecteerde gegevensbestanden onderzocht hoe de resultaten die met behulp van het nieuwe algoritme zijn verkregen, zich verhouden tot de resultaten die met andere, meer conventionele inductieve leertechnieken zijn behaald. Daartoe is een experiment uitgevoerd met tweeklassenproblemen waarin voor elk van de bestanden met verschillende methoden een classificator is geconstrueerd en als optimaliseringscriterium maximalisatie van de nauwkeurigheid is gebruikt. In een laatste experiment tenslotte, is voor één van de bestanden een andere diagnostische prestatiemaat als optimaliseringscriterium gebruikt, om aan te geven dat de meeste conventionele inductietechnieken op dit punt tekortschieten.

4.4.1 Maximalisatie van de classificatienauwkeurigheid

Eerder is aangegeven datvrijwel alle conventionele inductietechnieken een classificator met een zo hoog mogelijke nauwkeurigheid te trachten construeren. In experiment #4 worden de bestanden BU, GL2, HE en PI zowel aan een analyse door EXPLORE als door CART onderworpen. De resulterende classificatoren en hun prestaties worden vervolgens met elkaar en, waar mogelijk, met uit de literatuur bekende resultaten vergeleken.

[1]De eerste heuristiek zou dus ook hier geen afbreuk aan de prestaties hebben gedaan.

Werkwijze voor experiment #4

De CART-analyse is uitgevoerd met de standaardinstellingen van het algoritme. Dit houdt in dat: (a) de Gini-index als onzuiverheidsmaat wordt gebruikt om beslisknopen te splitsen; (b) 10-voudige *cross-validation* wordt gebruikt om de prestaties te schatten; (c) de *a priori* kansen van de twee klassen gelijk zijn aan de respectieve fracties in het bestand; (d) de misclassificatiekosten gelijk aan 1 zijn; (e) de uiteindelijke beslisboom op basis van de *'1 SE rule'* wordt geselecteerd; en (f) in een beslisknoop geen combinatie van kenmerken is toegestaan.

Bij de analyse met behulp van het EXPLORE-algoritme worden aan de prestatiematen sensitiviteit, specificiteit, positief predictieve waarde en negatief predictieve waarde van de resulterende classificator geen restricties opgelegd. Overeenkomstig de sub 3.3.5 omschreven werkwijze wordt: (a) eveneens 10-voudige *cross-validation* gebruikt om de prestaties te schatten; (b) de regellengte waarvoor de geschatte prestatiemaat maximaal is als optimale lengte beschouwd; en (c) de uiteindelijke beslisregel op basis van het hele bestand geconstrueerd. In een eerste stap van het experiment staat de optimaliteit van de oplossing voorop, waardoor met EXPLORE uitputtend is gezocht. Vervolgens zijn in een tweede stap van het experiment één of meer heuristieken gebruikt om de rekentijd te bekorten en is nagegaan in hoeverre afbreuk aan de optimaliteit van de oplossing wordt gedaan.

Resultaten van experiment #4

De *cross-validation* en resubstitutienauwkeurigheid van de resulterende classificatoren is voor beide algoritmen in tabel 4.5 weergegeven.

Tabel 4.5 Resultaten van experiment #4—Classificatienauwkeurigheid.

Bestand	CART		EXPLORE	
	Cross-validation	Resubstitutie	Cross-validation	Resubstitutie
BU	0.69275	0.69565	0.69252	0.69855
GL2	0.78528	0.80982	0.77833	0.81595
HE	0.83750*	0.83750*	0.89500	0.95000
PI	0.74349	0.77214	0.77030	0.77214

**A priori* waarschijnlijkheid van de meest voorkomende klasse.

Uit tabel 4.5 blijkt dat alle *cross-validation* prestaties van de beslisregels gegenereerd door EXPLORE vergelijkbaar zijn met, of beter zijn dan die van de beslisbomen gegenereerd door CART; slechts voor GL2 vindt CART een betere oplossing, waarbij gemiddeld genomen anderhalf object méér correct wordt geclassificeerd. De afwijking in de resubsti-

tutienauwkeurigheid duidt er op dat de uiteindelijk geselecteerde classificatoren verschillen. Om dit te illustreren zijn de resulterende classificatoren als beslisregels in tabel 4.6 weergegeven.

Tabel 4.6 Resultaten van experiment #4—Resulterende beslisregels.

Bestand	CART	EXPLORE
BU	(*Gammagt* > 20.5) OF (*Sgpt* ≤ 19.5)	(*Gammagt* > 21.5) OF (*Sgpt* ≤ 19.5)
GL2	(*Al* > 1.42) OF (*Ca* > 10.4)	(*Mg* ≤ 3.29) OF (*Ca* ≤ 8.32)
HE	–*	(*Alk Phosphate* > 236.5) OF
		(*Protime* ≤ 46.5) EN (*Histology* > 1.5)
PI	(*Glucose* > 127) EN (*Mass Index* > 29.9)	(*Glucose* > 127) EN (*Mass Index* > 29.9)

*Geen kenmerktests; meerderheidsklasse wordt bij verstek gekozen.

Tabel 4.6 laat zien dat voor BU en PI (nagenoeg) dezelfde beslisregels worden gevonden, terwijl voor GL2 en HE afwijkende kenmerktests worden gebruikt. Nader onderzoek van de resultaten van het GL2-probleem geeft aan dat volgens beide inductiemethoden het kenmerk *'Ca'* het meest relevant is, maar dat CART het kenmerk *'Al'* prefereert boven *'Mg'*. Ook voor HE blijkt uit de resultaten van de CART-analyse dat de kenmerken *'Protime'*, *'Alk Phosphate'* en *'Histology'* het meest relevant worden geacht. CART selecteert op basis van de *'1 SE rule'* (zie pagina 26) echter uiteindelijk de boom met één eindknoop, terwijl EXPLORE met de hiervoor genoemde kenmerken een aanzienlijk beter presterende regel construeert. Uit de CART-analyse blijkt tevens dat tijdens het inductieproces een beslisboom wordt gegenereerd die (nagenoeg) overeenkomt met de door EXPLORE afgeleide beslisregel, maar dat een *standard error* van 1 te groot blijkt te zijn om deze boom als beste te selecteren. Heuristieken om de uiteindelijke classificator te bepalen zijn derhalve nog een punt van nader onderzoek.

Vergelijking met de uit de literatuur bekende, in tabel B.1 op pagina 84 weergegeven prestaties bevestigt dat door EXPLORE vergelijkbare of betere beslisregels zijn gevonden.

Versnelling van het inductieproces

Voorts dient te worden opgemerkt dat het construeren van beslisbomen aanzienlijk sneller verloopt dan het construeren van optimale beslisregels. Zo vergt het construeren van een beslisboom voor het PI-probleem met CART circa een halve minuut rekentijd op een HP 715/80 werkstation, terwijl het uitputtend zoeken naar optimale beslisregel met EXPLORE circa 20 uur (!) in beslag neemt. Dit laatste blijkt vooral te wijten aan de 10-voudige *cross-validation* techniek, omdat voor 10 afzonderlijke leerverzamelingen bij toenemende regellengte een uitputtend inductieproces wordt uitgevoerd. Een mogelijke

oplossing hiervoor kan zijn heuristieken te gebruiken of het leerproces te parallelliseren.

Het gebruik van de eerste heuristiek (een kenmerk mag hoogstens één maal per beslisregel voorkomen) en de tweede heuristiek (kies één rekenkundige operator op basis van de oppervlakte W onder de ROC-curve) resulteert in een aanzienlijk sneller leerproces en blijkt voor de vier geanalyseerde gegevensbestanden geen effect op de eindoplossing te hebben. Geconcludeerd kan worden dat deze heuristieken op grond van de empirische resultaten voor BU, GL2, HE en PI zeer waardevol zijn gebleken.

4.4.2 Optimalisatie van de diagnostische prestaties

Alhoewel maximalisatie van de classificatienauwkeurigheid een veelgebruikt optimaliseringscriterium is, zijn voor medisch-diagnostische toepassingen prestatiematen als sensitiviteit en specificiteit veelal relevanter. Met uitzondering van het PVM-algoritme zijn conventionele inductietechnieken niet in staat om een classificator te construeren die optimaal is onder voorwaarden die aan de diagnostische prestaties van de resulterende classificator zijn opgelegd. In de praktijk is het optimaliseren van een classificator daarom veelal een *trial-and-error* proces. Voor CART bijvoorbeeld, is in [Breiman *et al.*, 1984, pagina 185] aangegeven dat de diagnostische prestaties van de resulterende classificator kunnen worden beïnvloed door verschillende misclassificatiekosten in rekening te brengen. De precieze relatie tussen veranderingen in deze kosten en de veranderingen in de diagnostische prestaties is doorgaans echter onduidelijk, waardoor men gedwongen wordt de classificator langs een ondoorzichtige en omslachtige weg te optimaliseren.

Met de nieuwe methode is het mogelijk één van de diagnostische prestatiematen te maximaliseren, onder minimum vereiste waarden waaraan de resterende maten moeten voldoen. Omdat voor PI in experiment #4 door CART en EXPLORE dezelfd beslisregel is gevonden, is het interessant dit bestand in experiment #5 te gebruiken om een nieuwe classificator te construeren, waarbij als optimaliseringscriterium maximalisatie van de nauwkeurigheid is gebruikt.

Werkwijze voor experiment #5

Uit experiment #4 blijkt dat de regel

als (*Glucose* > 127) EN (*Mass Index* > 29.9) **dan** 'Diabetes Mellitus'

die CART presenteert, een maximale *cross-validation* nauwkeurigheid van 0.74349 heeft. De bijbehorende sensitiviteit en specificiteit zijn respectievelijk gelijk aan 0.48508 en 0.88200. De sensitiviteit van deze regel als classificator voor de aanwezigheid van de ziekte *diabetes mellitus* laat echter sterk te wensen over wanneer het van belang is zo veel mogelijk zieken daadwerkelijk als te 'ziek' te classificeren, maar dat niet ten koste van de huidige nauwkeurigheid mag gaan. Immers wordt in slechts 48.5% van de gevallen de aanwezigheid van de ziekte terecht vastgesteld, terwijl in 11.8% van de

gevallen de aanwezigheid daarvan ten onrechte wordt vastgesteld. Zodoende wordt de sensitiviteit van de resulterende classificator gemaximaliseerd onder de voorwaarde dat de nauwkeurigheid ten minste gelijk aan 0.74349 blijft.

Resultaten van experiment #5

De *cross-validation* en resubstitutieprestaties van zowel de eerder gevonden, als de nieuwe beslisregel zijn in tabel 4.7 weergegeven.

Tabel 4.7 Resultaten van experiment #5—Prestatiematen van beslisregels.

	Eerdere regel (CART)		Nieuwe regel (EXPLORE)	
	Cross-validation	Resubstitutie	Cross-validation	Resubstitutie
Nauwkeurigheid	0.74349	0.77214	0.74545	0.74609
Sensitiviteit	0.48508	0.55970	0.62678	0.65672
Specificiteit	0.88200	0.88600	0.80906	0.79400

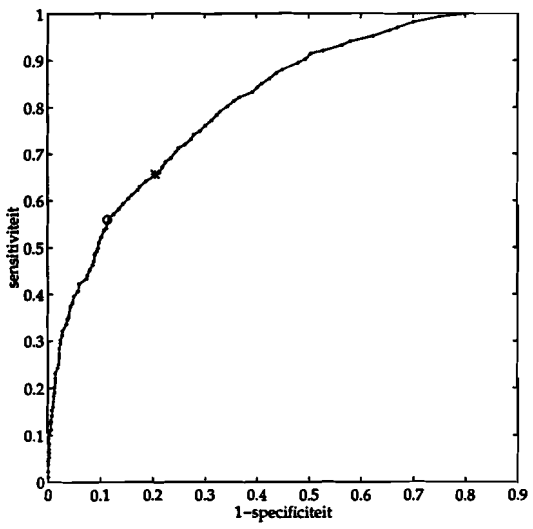
Uit tabel 4.7 blijkt dat er een door EXPLORE gegenereerde beslisregel bestaat waarvoor de *cross-validation* nauwkeurigheid nagenoeg hetzelfde is, terwijl de sensitiviteit beduidend hoger is. Uiteraard gaat de toename in sensitiviteit ($\approx$ 14%) ten koste van een afname in specificiteit ($\approx$ 7%). Of de uitwisseling van correct-positieve tegen correct-negatieve beslissingen zinvol is, is niet zonder meer te zeggen, omdat dit afhankelijk is van de prevalentie en de specifieke doelstelling van het classificatieprobleem.

Veel belangrijker is echter dat hiermee is aangetoond dat het mogelijk is de diagnostische prestaties van de resulterende classificator aan te passen aan de specifieke voorwaarden die de gebruiker oplegt.

Constructie van probleem-specifieke classificatoren

Doordat de sensitiviteit van de resulterende beslisregel kan worden gestuurd, is het mogelijk probleem-specifieke classificatoren te construeren en, afhankelijk van de toepassing, een selectie te maken.

Als voorbeeld is voor PI de minimaal vereiste resubstitutiesensitiviteit in stappen van 1% gevarieerd, waarbij telkens een nieuwe classificator is geconstueerd door als optimaliseringscriterium maximalisatie van de specificiteit (of nauwkeurigheid) te gebruiken. Zodoende ontstaat een verzameling probleem-specifieke classificatoren waarvan de prestaties in de ROC-curve van figuur 4.6 op pagina 57 zijn afgebeeld.



Figuur 4.6 Empirische ROC-curve voor PI.

De door CART geïnduceerde beslisregel met de hoogste nauwkeurigheid,

als (*Glucose* > 127) EN (*Mass Index* > 29.9) **dan** 'Diabetes Mellitus', (4.3)

is in figuur 4.6 als punt (∘) weergegeven. De door EXPLORE op sensitiviteit geoptimaliseerde beslisregel,

als (*Glucose* > 123) EN (*Mass Index* > 26.3) **dan** 'Diabetes Mellitus', (4.4)

is in figuur 4.6 als punt (∗) weergegeven.

Opgemerkt wordt dat het patroon van kenmerken en rekenkundige operatoren van de verzameling beslisregels die in figuur 4.6 als punt (·) zijn weergegeven, zeer frequent overeenstemt met dat van de regels (4.3) en (4.4), maar dat de beslisdrempels anders zijn ingesteld. Klaarblijkelijk zijn de kenmerken '*Glucose*' en '*Mass Index*' voor het PI-probleem zeer geschikt om er optimale beslisregels mee te construeren.

Wanneer voor de afzonderlijke kenmerktests (*Glucose* > ?) en (*Mass Index* > ?) ROC-curven worden geconstrueerd, dan blijkt op grond van de oppervlakken onder die curven dat deze kenmerken univariaat reeds als veelbelovend kunnen worden aangemerkt. Een punt van nader onderzoek zou daarom kunnen zijn na te gaan in hoeverre ROC-curven kunnen bijdragen aan het selecteren van veelbelovende kenmerken en het instellen van beslisdrempels wanneer kenmerktests worden gecombineerd.

Afhankelijkheid randvoorwaarden

De in experiment #5 vereiste minimale nauwkeurigheid is gelijk aan de eerder in experiment #4 door CART geschatte *cross-validation* nauwkeurigheid. Echter, omdat door CART en EXPLORE in experiment #5 dezelfde regel is gevonden, had ook de door EXPLORE geschatte nauwkeurigheid (0.77030) gebruikt kunnen worden om de sensitiviteit te optimaliseren. Bij herhaling van experiment #5 met deze waarde, blijkt dat voor een aantal leerverzamelingen van het *cross-validation* experiment geen classificator kan worden geconstrueerd, waardoor een 10-voudige schatting van de sensitiviteit niet mogelijk is. Een ander punt van nader onderzoek zou kunnen zijn na te gaan hoe in een dergelijke situatie een goede schatting van de werkelijke prestaties kan worden verkregen.

Ook zij hiermee aangegeven dat bij het gebruik van de methode enige omzichtigheid dient te worden betracht. De diagnostische prestatiematen zijn namelijk afhankelijk, waardoor de tevoren opgegeven minimumwaarden zorgvuldig dienen te worden gekozen. In dit verband wordt opgemerkt dat wanneer een willekeurige minimumwaarde voor sensitiviteit of specificiteit is vereist en aan de overige prestatiematen geen restricties zijn opgelegd, het altijd mogelijk is een classificator te construeren die aan de voorwaarden voldoet. Voor de predictieve waarden en de nauwkeurigheid hoeft dit echter niet het geval te zijn. Ook bij specificatie van meerdere minimumwaarden dient er rekening mee te worden gehouden dat er geen oplossing hoeft te bestaan.

4.5 CONCLUSIES

De experimenten laten zien dat de eerste heuristiek, waarmee een kenmerk ten hoogste één maal per beslisregel kan voorkomen, voor de geanalyseerde bestanden geen effect heeft op de resulterende classificator. Van de tweede heuristiek, die *a priori* één rekenkundige operator per kenmerk bepaalt, is aangetoond dat deze, afhankelijk van de methode waarop zij is gebaseerd, de oplossing wel nadelig kan beïnvloeden. De resultaten laten tevens zien dat de derde heuristiek, die het aantal potentiële drempelwaarden reduceert, voor een minimum aantal objecten van 5–10% van het totaal aan de optimaliteit nauwelijks of geen afbreuk doet, terwijl de complexiteitsreductie aanzienlijk kan zijn.

Vergeleken met resultaten van andere, meer conventionele inductietechnieken blijkt dat, wanneer als optimaliseringscriterium maximalisatie van de classificatienauwkeurigheid wordt gebruikt, vergelijkbare of betere classificatoren kunnen worden geconstrueerd.

De resultaten bevestigen het uit de literatuur bekende fenomeen dat relatief korte regels vaak tot een betere oplossing leiden dan een verzameling ingewikkelder regels [Michalski *et al.*, 1986; Quinlan, 1987a; Weiss *et al.*, 1990; Holte, 1993]. Bovendien wordt aangetoond dat als een goede oplossing in de vorm van een korte regel bestaat, de nieuwe methode veelal in het voordeel is in vergelijking met andere inductietechnieken.

Voorts is in vergelijking met technieken voor het induceren van beslisbomen een duidelijke uitwisseling tussen snelheid en optimaliteit waar te nemen. De hiërarchische benadering die kenmerkend is voor beslisbomen, resulteert in een sneller leerproces, terwijl volgens de nieuwe benadering het uitvoeriger combineren van kenmerktests de optimaliteit van de oplossing ten goede komt.

Tenslotte is aangetoond dat, in tegenstelling tot de meeste andere technieken, het met de nieuwe methode mogelijk is één van de diagnostische prestatiematen te optimaliseren onder oplegging van minimum vereiste waarden aan de overige prestatiematen.

HOOFDSTUK 5

Conclusies

Uit literatuuronderzoek blijkt dat vrijwel alle conventionele inductietechnieken om twee redenen niet geschikt zijn een classificator te construeren die optimaal is onder voorwaarden die aan de diagnostische prestatiematen van de resulterende classificator zijn opgelegd. De eerste reden is dat de (diagnostische) prestaties tijdens het inductieproces moeilijk of niet te sturen zijn, doordat de relatie tussen heuristische evaluatiefuncties die bij dit proces worden gebruikt en de te optimaliseren prestatiemaat onduidelijk is. De tweede reden heeft betrekking op de optimaliteit. Doordat het stapsgewijs verlopende inductieproces lokaal optimale beslissingen neemt en er geen *backtracking*-mechanisme aanwezig is om eerdere beslissingen te herzien, kan globale optimaliteit van de resulterende classificator niet gegarandeerd worden.

Het PVM-algoritme van Weiss *et al.* (1987) tracht deze bezwaren te ondervangen door met behulp van een heuristische zoekprocedure 'veelbelovende' beslisregels te genereren. Kenmerktests worden hierbij uitvoeriger gecombineerd en voor meerdere drempelwaardencombinaties geëvalueerd. Eén van de vijf diagnostische prestatiematen sensitiviteit, specificiteit, positief predictieve waarde, negatief predictieve waarde of nauwkeurigheid kan worden geoptimaliseerd onder oplegging van minimum vereiste waarden aan de resterende prestatiematen.

Een benadering in de richting van meer uitgebreide zoektechnieken lijkt derhalve vooruitzichten te bieden. Om hierin meer inzicht te verkrijgen is nagegaan in hoeverre het uitputtend genereren van alle mogelijke beslisboom- en beslisregelclassificatoren mogelijk is. Uit analyses van de rekenkundige complexiteit daarvan blijkt dat het genereren

van beslisregels efficiënter is. Er is een nieuwe methode ontwikkeld voor het uitputtend genereren van beslisregels die optimaal zijn onder oplegging van restricties aan de diagnostische prestatiematen. Bij toepassing in de praktijk blijkt het echter vaak nodig te zijn heuristieken te gebruiken om de omvang van de zoekruimte tot aanvaardbare proporties terug te brengen. Er is een aantal eenvoudige heuristieken voorgesteld en geïmplementeerd. De nieuwe methode lijkt daarmee op die van PVM, met dien verstande dat er een diepgaander analyse aan ten grondslag ligt, andere heuristieken zijn geïntroduceerd en het mogelijk is al naar gelang de complexiteit van de classificatietaak uitputtend te zoeken of één of meer heuristieken te gebruiken.

Uit experimenten met uit de literatuur bekende gegevensbestanden blijkt dat een aantal 'veelbelovende' heuristieken nauwelijks of geen afbreuk aan de optimaliteit van de oplossing doet, terwijl de rekenkundige complexiteit aanzienlijk wordt gereduceerd. Vergeleken met resultaten van andere, meer conventionele inductietechnieken blijkt dat, wanneer als optimaliseringscriterium maximalisatie van de nauwkeurigheid wordt gebruikt, er vergelijkbare of betere classificatoren worden geconstrueerd.

Voorts blijkt uit een vergelijking van EXPLORE met het CART-algoritme voor inductie van beslisbomen, dat er een duidelijke uitwisseling tussen snelheid en optimaliteit bestaat. De hiërarchische benadering, die kenmerkend is voor beslisbomen, resulteert in een sneller leerproces, terwijl volgens de nieuwe benadering het uitvoeriger combineren van kenmerktests de optimaliteit van de oplossing ten goede komt.

Met betrekking tot de praktische bruikbaarheid van de nieuwe methode wordt geconcludeerd dat:

- EXPLORE momenteel beperkt is tot het induceren van korte regels voor relatief eenvoudige, laagdimensionale classificatietaken;
- het vorige punt vaak geen beperking hoeft te vormen, omdat het door empirische resultaten in de literatuur bevestigd is dat relatief korte regels vaak tot een betere oplossing leiden dan een verzameling ingewikkelder regels; en
- als een goede oplossing in de vorm van een korte regel bestaat, EXPLORE veelal in het voordeel is.

Aangetoond is dat, in tegenstelling tot de meeste andere inductietechnieken, het met de nieuwe methode mogelijk is de diagnostische prestaties van de resulterende classificator aan te passen aan de specifieke voorwaarden die de gebruiker oplegt. Zodoende wordt een mogelijkheid geboden om voor specifieke classificatietaken optimale beslisregels te construeren.

HOOFDSTUK 6

Aanbevelingen

Eerder in dit rapport zijn op diverse plaatsen in hoofdstuk 3 en 4 suggesties aangegeven om het inductieproces te versnellen. Daarnaast zijn naar aanleiding van de experimenten een aantal vragen gerezen die onbeantwoord zijn gebleven. In dit hoofdstuk worden de suggesties en vragen die verdere aandacht verdienen, op een rijtje gezet.

Versnelling van het leerproces

Met de nieuwe methode is het mogelijk om al naar gelang de complexiteit van de classificatietaak uitputtend te zoeken of één of meer heuristieken te gebruiken.

Zoals sub 3.3.4 is aangegeven, kan de complexiteit van het uitputtend zoeken verder worden gereduceerd wanneer bij het genereren van expressies, equivalenten buiten beschouwing worden gelaten. Zo bleek het mogelijk om het aantal k-conjuncties van n^k tot $\binom{n+k-1}{k}$ te reduceren. Een algoritme om dit te realiseren, kan direct worden afgeleid uit de manier waarop het aantal logisch equivalente k-conjuncties in bijlage A.3 is berekend. Een manier om op grond hiervan enkel unieke DNF-expressies te genereren, is echter niet aangegeven, en zou een onderwerp van nader onderzoek kunnen zijn.

Eveneens is sub 3.3.4 aangegeven dat het aantal operatorcombinaties verder kan worden teruggebracht dan met de nu gebruikte heuristieken. De wijze waarop dit zou kunnen plaatsvinden is tevens weergegeven. Toch is een diepgaander onderzoek naar een methode voor het genereren van zinvolle regels aanbevelenswaardig.

Nieuwe heuristieken

Naast deze maatregelen om de complexiteit van het uitputtend zoeken verder terug te dringen, zouden ook nieuwe heuristieken kunnen worden toegepast. Daarbij verdient het aanbeveling de op pagina 51 omschreven meerstapsanalyse voor het bepalen van 'interessante' drempelwaarden te implementeren en te evalueren.

Empirische resultaten hebben aangetoond dat veelbelovende langere regels vaak door combinatie van kortere regels worden verkregen. De suggestie om een tabel van veelbelovende regels bij te houden, zou derhalve ook verder onderzocht moeten worden.

Ook het gebruik van ROC-curven om veelbelovende kenmerken te selecteren en beslisdrempels in te stellen wanneer kenmerktests worden gecombineerd, lijkt op grond van de empirische resultaten interessante vooruitzichten te bieden. Onderzocht zou kunnen worden in hoeverre ROC-curven hieraan een zinvolle bijdrage kunnen leveren.

Gevoeligheidsanalyse

Naast de hiervoor genoemde overwegingen en maatregelen, verdient het aanbeveling de stabiliteit van door EXPLORE geconstrueerde regels te onderzoeken. Een op *cross-validation* gebaseerde gevoeligheidsanalyse zou hierin enig inzicht kunnen verschaffen. In plaats van de enkele beste regel van optimale lengte te selecteren, zou bijvoorbeeld ook een modale regel, d.w.z. een regel die relatief frequent voorkomt, of een modaal patroon van kenmerken en rekenkundige operatoren kunnen worden geselecteerd. Nader onderzocht dient te worden hoe alternatieve gevoeligheidsanalyses kunnen worden gebruikt om de stabiliteit te quantificeren. Speciale aandacht verdient het na te gaan welke strategieën gevolgd kunnen worden wanneer blijkt dat de variabiliteit in regels of patronen relatief is.

Ook voor het geval dat het niet mogelijk blijkt te zijn om op basis van één of meer (*cross-validation*) leerbestanden een classificator te construeren die aan de restricties voldoet, dient te worden onderzocht of op een alternatieve wijze een goede schatting van de werkelijke prestaties kan worden bepaald.

Strategieën voor meerklassenproblemen

Tenslotte zij opgemerkt dat het wenselijk is de methode niet alleen voor twee-, maar ook voor meerklassenproblemen te kunnen gebruiken. Een mogelijke eerste strategie voor gebruik bij meerklassenproblemen is reeds bij experiment #1 (sub 4.3.1) aan de orde gesteld. Alternatieve strategieën zouden in kaart dienen te worden gebracht en te worden geëvalueerd op de uit de literatuur bekende gegevensbestanden. Tevens dient te worden onderzocht hoe begrippen als 'sensitiviteit' en 'specificiteit' kunnen worden gegeneraliseerd.

Bibliografie

Andrews, G. (1976). The theory of partitions. In Rota, G., editor, *Encyclopedia of Mathematics and Its Applications*, pages 214–216. Addison-Wesley Publishing Company.

Argentiero, P., Chin, R., & Beaudet, P. (1982). An automated approach to the design of decision tree classifiers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-4(1): 51–57.

Bamber, D. (1975). The area above the ordinal dominance graph and the area below the receiver operating graph. *Journal of Mathematical Psychology*, 12: 387–415.

Beck, J. & Shultz, E. (1986). The use of relative operating characteristic (ROC) curves in test performance evaluation. *Archives of Pathology and Laboratory Medicine*, 110: 13–20.

Breiman, L., Friedman, J., Olshen, R., & Stone, C. (1984). *Classification and Regression Trees*. Wadsworth Int., Belmont, CA.

Buntine, W. (1989). Learning classification rules using Bayes. In Segre, A., editor, *Proceedings of the Sixth International Workshop on Machine Learning*, pages 94–98. Morgan Kaufmann, San Mateo, CA.

Buntine, W. & Niblett, T. (1992). A further comparison of splitting rules for decision-tree induction. *Machine Learning*, 8(1): 75–86.

Cestnik, B., Kononenko, I., & Bratko, I. (1987). ASSISTANT86: A knowledge elicitation tool for sophisticated users. In Bratko, I. & Lavrač, N., editors, *Progress in Machine Learning*, pages 31–45. Sigma Press, Wilmslow.

Clancey, W. (1985). Heuristic classification. *Artificial Intelligence*, 27: 289–350.

Clark, P. & Boswell, R. (1991). Rule induction with CN2: Some recent improvements. In Kodratoff, Y., editor, *Machine Learning—EWSL-91*, pages 151–163. Springer-Verlag.

Clark, P. & Niblett, T. (1989). The CN2 induction algorithm. *Machine Learning*, 3(4): 261–283.

Connell, F. & Koepsell, T. (1985). Measures of gain in certainty from a diagnostic test. *American Journal of Epidemiology*, 121(5): 744–753.

Diaconis, P. & Efron, B. (1983). Computer-intensive methods in statistics. *Scientific American*, 248.

Duda, R. & Hart, P. (1973). *Pattern Classification and Scene Analysis*. Wiley-Interscience, New York.

Efron, B. & Gong, G. (1983). A leisurely look at the bootstrap, the jackknife and cross-validation. *American Statistician*, 37(1): 36–48.

Evett, I. & Spiehler, E. (1987). Rule induction in forensic science. Technical report, Central Research Establishment, Home Office Forensic Science Service, Berkshire.

Fayyad, U. & Irani, K. (1992). On the handling of continuous-valued attributes in decision tree generation. *Machine Learning*, 8(1): 87–102.

Fisher, R. (1936). The use of multiple measurements in taxonomic problems. *Annual Eugenics*, 7: 179–188.

Flegg, H. (1965). *Boolean Algebra and its Application*. Blackie, London.

Friedman, J. (1977). A recursive partitioning decision rule for nonparametric classification. *IEEE Transactions on Computers*, C-26: 404–408.

Fukunaga, K. (1972). *Introduction to Statistical Pattern Recognition*. Academic Press, New York.

Galen, R. & Gambino, S. (1975). *Beyond Normality: The Predictive Value and Efficiency of Medical Diagnoses*. Wiley, New York.

Gleser, M. & Collen, M. (1972). Towards automated medical decisions. *Computers and Biomedical Research*, 5: 180–189.

Green, D. & Swets, J. (1966). *Signal Detection Theory and Psychophysics*. Wiley, New York.

Hart, A. (1985). Experience in the use of an inductive system in knowledge engineering. In Bramer, M., editor, *Research and Development in Expert Systems*. Cambridge University Press, Cambridge.

Henrichson, E. & Fu, K. (1969). A nonparametric partitioning procedure for pattern classification. *IEEE Transactions on Computers*, C-18(7): 614–624.

Highleyman, W. (1962). The design and analysis of pattern recognition experiments. *Bell Systems Technical Journal*, 41: 723–744.

Holte, R. (1993). Very simple classification rules perform well on most commonly used datasets. *Machine Learning*, 11(1): 63–90.

James, M. (1985). *Classification Algorithms*. Collins, London.

Kern, J., Deželić, G., Dürrigl, T., & Vuletić, S. (1993). Medical decision making based on inductive learning method. *Artificial Intelligence in Medicine*, 5: 213–223.

Lachenbruch, P. & Mickey, M. (1968). Estimation of error rates in discriminant analysis. *Technometrics*, 10: 1–11.

Landeweerd, G., Timmers, T., Gelsema, E., Bins, M., & Halie, M. (1983). Binary tree versus single level tree classification of white blood cells. *Pattern Recognition*, 16(6): 571–577.

Lin, Y. & Fu, K. (1983). Automatic classification of cervical cells using a binary tree classifier. *Pattern Recognition*, 16(1): 69–80.

Magnus, W., Oberhettinger, F., & Soni, R. (1966). *Formulas and Theorems for the Special Functions of Mathematical Physics*. Springer, Berlin, 3rd edition.

Meisel, W. & Michalopoulos, D. (1973). A partitioning algorithm with application in pattern classification and optimization of decision trees. *IEEE Transactions on Computers*, C-22(1): 93–103.

Michalski, R., Mozetič, I., Hong, J., & Lavrač, N. (1986). The multi-purpose incremental learning system AQ15 and its testing application to three medical domains. In *Proceedings of the Fifth National Conference on Artificial Intelligence, AAAI-86*, pages 1041–1045, Los Altos, CA. Morgan Kaufmann Publishers, Inc.

Mingers, J. (1989a). An empirical comparison of pruning methods for decision tree induction. *Machine Learning*, 4(2): 227–243.

Mingers, J. (1989b). An empirical comparison of selection measures for decision-tree induction. *Machine Learning*, 3(4): 319–342.

Mui, J. & Fu, K. (1980). Automated classification of nucleated blood cells using a binary tree classifier. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-2(5): 429–443.

Murphy, P. & Aha, D. (1994). UCI repository of machine learning databases. [machine-readable data repository], Department of Information and Computer Science, University of California, Irvine, CA.

Narenda, P. & Fukunaga, K. (1977). A branch and bound algorithm for feature subset selection. *IEEE Transactions on Computers*, C-26: 917–922.

Payne, H. & Meisel, W. (1977). An algorithm for constructing optimal binary decision trees. *IEEE Transactions on Computers*, C-26: 905–916.

Quinlan, J. (1986). Induction of decision trees. *Machine Learning*, 1(1): 81–106.

Quinlan, J. (1987a). Generating production rules from decision trees. In Joshi, A., editor, *Proceedings of the Tenth International Joint Conference on Artificial Intelligence, IJCAI-87*, pages 304–307, Los Altos. Morgan Kaufmann Publishers, Inc.

Quinlan, J. (1987b). Simplifying decision trees. *International Journal of Man-Machine Studies*, 27: 221–234.

Quinlan, J. (1990). Decision trees and decisionmaking. *IEEE Transactions on Systems Man and Cybernetics*, SMC-20(2): 339–346.

Rounds, E. (1980). A combined non-parametric approach to feature selection and binary decision tree design. *Pattern Recognition*, 12: 313–317.

Sethi, I. & Sarvarayudu, G. (1982). Hierarchical classifier design using mutual information. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-4(4): 441–445.

Smith, J., Everhart, J., Knowler, W., & Johannes, R. (1988). Using the ADAP learning algorithm to forecast the onset of diabetes mellitus. In *Proceedings of the Symposium on Computer Applications and Medical Care*, pages 261–265. IEEE Computer Society Press.

Sokal, R. & Rohlf, F. (1981). *Biometry*. Freeman, San Francisco.

Stone, M. (1974). Cross-validatory choice and assessment of statistical predictions. *Journal of the Royal Statistical Society, Series B*, 36(2): 111–147.

Swain, P. & Hauska, H. (1977). The decision tree classifier: design and potential. *IEEE Transactions on Geoscience Electronics*, GE-15(3): 142–147.

Talmon, J. (1986). A multiclass nonparametric partitioning algorithm. *Pattern Recognition Letters*, 4(1): 31–38.

Toussaint, G. (1974). Bibliography on estimation of misclassification. *IEEE Transactions on Information Theory*, IT-20(4): 472–479.

Wang, Q. & Suen, C. (1984). Analysis and design of a decision tree based on entropy reduction and its application to large character set recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-6(4): 406–417.

Weiss, S., Galen, R., & Tadepalli, P. (1987). Optimizing the predictive value of diagnostic decision rules. In *Proceedings of the Sixth National Conference on Artificial Intelligence, AAAI-87*, pages 521–526, Los Altos, CA. Morgan Kaufmann Publishers, Inc.

Weiss, S., Galen, R., & Tadepalli, P. (1990). Maximizing the predictive value of production rules. *Artificial Intelligence*, 45(1–2): 47–71.

Weiss, S. & Kapouleas, I. (1989). An empirical comparison of pattern recognition, neural nets, and machine learning classification methods. In Sridharan, N., editor, *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence, IJCAI-89*, pages 781–787, San Mateo, CA. Morgan Kaufmann.

Weiss, S. & Kulikowski, C. (1990). *Computer Systems That Learn: Classification and Prediction Methods from Statistics, Neural Nets, Machine Learning and Expert Systems*. Morgan Kaufmann Publishers, Inc., San Mateo, CA.

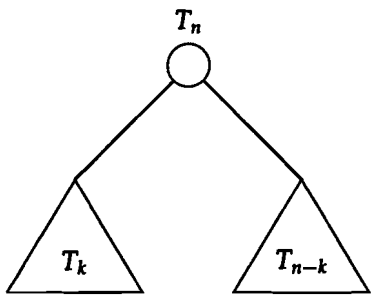
BIJLAGE A

Complexiteitsanalyse

A.1 TOPOLOGIEËN VAN BINAIRE BOMEN

In deze paragraaf volgt een analyse van het aantal verschillende topologieën van binaire bomen met n eindknopen of bladeren.

Laat daartoe T_n een binaire boom met n bladeren zijn en laat het aantal topologisch verschillende bomen door a_n worden genoteerd.



Figuur A.1 Een boom T_n met $n, n \geq 2$, bladeren.

Veronderstel dat, zoals is weergegeven in figuur A.1, voor $n \geq 2$ de linkertak uitgaande van de wortel van T_n een subboom T_k is met k, $1 \leq k < n$, bladeren, dan is de rechtertak uitgaande van de wortel van T_n een subboom T_{n-k} met $n - k$ bladeren.

Laat het aantal subbomen T_k en T_{n-k} respectievelijk door a_k en a_{n-k} worden genoteerd.

Het aantal mogelijke bomen bij gegeven k bedraagt dan $a_k a_{n-k}$. Het totaal aantal verschillende boomtopologieën wordt verkregen door over alle mogelijke waarden van k te sommeren:

$$a_n = \sum_{k=1}^{n-1} a_k a_{n-k}, \qquad n \geq 2, \tag{A.1}$$

waarbij voor de triviale boom T_1 geldt dat $a_1 = 1$. Indien voor het aantal lege bomen T_0 per definitie geldt dat $a_0 = 0$, dan kan vergelijking (A.1) worden geschreven als

$$a_n = \sum_{k=0}^{n} a_k a_{n-k}, \qquad n \geq 2. \tag{A.2}$$

De recurrente betrekking van vergelijking (A.2) kan met behulp van z-transformatie worden opgelost.

De unilateraal z-getransformeerde van de rij a_n is gedefinieerd als

$$A_I(z) := \sum_{n=0}^{\infty} a_n z^{-n}. \tag{A.3}$$

Substitutie van vergelijking (A.2) geeft

$$\begin{aligned} A_I(z) &= z^{-1} + \sum_{n=2}^{\infty} \left(\sum_{k=0}^{n} a_k a_{n-k} \right) z^{-n} \\ &= z^{-1} + \sum_{n=0}^{\infty} \left(\sum_{k=0}^{n} a_k a_{n-k} \right) z^{-n} \\ &= z^{-1} + \left(\sum_{n=0}^{\infty} a_n z^{-n} \right)^2 \\ &= z^{-1} + A_I^2(z). \end{aligned} \tag{A.4}$$

Oplossing van het tweedegraads polynoom in $A_I(z)$ van vergelijking (A.4) levert de wortels

$$\begin{aligned} A_I(z)_{1,2} &= \frac{1 \pm \sqrt{1 - 4z^{-1}}}{2} \\ &= \frac{1}{2} \pm \frac{1}{2}(1 - 4z^{-1})^{\frac{1}{2}}. \end{aligned} \tag{A.5}$$

Newton's Binomiumformule voor reële α,

$$(1 + x)^{\alpha} := \sum_{n=0}^{\infty} \binom{\alpha}{n} x^n,$$

met daarin

$$\binom{\alpha}{n} := \frac{\alpha(\alpha - 1) \cdots (\alpha - n + 1)}{n(n-1) \cdots 1},$$

levert na substitutie van $x = -\beta z^{-1}$

$$(1-\beta z^{-1})^{\alpha} = \sum_{n=0}^{\infty} \binom{\alpha}{n} (-\beta)^n z^{-n}. \tag{A.6}$$

Volgens (A.3) is het rechterlid van vergelijking (A.6) per definitie gelijk aan de unilateraal z-getransformeerde van $(-\beta)^n \binom{\alpha}{n}$.

Hieruit volgt door inverse z-transformatie van (A.5) dat voor $n \geq 0$

$$a_{n,1} = \frac{1}{2}\delta_n + \frac{1}{2}(-4)^n \binom{\frac{1}{2}}{n}, \tag{A.7}$$

$$a_{n,2} = \frac{1}{2}\delta_n - \frac{1}{2}(-4)^n \binom{\frac{1}{2}}{n}, \tag{A.8}$$

waarin

$$\delta_n = \begin{cases} 1 & n = 0, \\ 0 & n \neq 0 \end{cases}.$$

Door substitutie van $n = 0$ in vergelijking (A.7) en (A.8) volgt dat $a_{0,1} \neq a_0$, waardoor oplossing $a_{n,1}$ vervalt en dus

$$a_n = a_{n,2} = \frac{1}{2}\delta_n - \frac{1}{2}(-4)^n \binom{\frac{1}{2}}{n}, \qquad n \geq 0. \tag{A.9}$$

Vergelijking (A.9) kan voor $n \geq 1$ verder worden vereenvoudigd

$$\begin{aligned} a_n &= -\frac{1}{2}(-4)^n \binom{\frac{1}{2}}{n} \\ &= (-1)^{n+1} 2^{2n-1} \frac{\frac{1}{2}(-\frac{1}{2})(-\frac{3}{2})\cdots(\frac{3}{2}-n)}{n(n-1)\cdots 1} \\ &= 2^{2n-1} \frac{\frac{1}{2^n} 1(1)(3)\cdots(2n-3)}{n(n-1)\cdots 1} \\ &= 2^{n-1} \frac{(2n-2)!}{n!(2)(4)\cdots(2n-2)} \\ &= \frac{1}{n} \frac{(2n-2)!}{(n-1)!(n-1)!} \end{aligned}$$

tot

$$a_n = \frac{1}{n}\binom{2(n-1)}{n-1}, \qquad n \geq 1. \tag{A.10}$$

Het aantal verschillende topologieën van binaire bomen met n beslisknopen is derhalve gelijk aan

$$a_{n+1} = \frac{1}{n+1}\binom{2n}{n}, \qquad n \geq 0. \tag{A.11}$$

A.2 GELABELDE TOPOLOGIEËN VAN BINAIRE BOMEN

In deze paragraaf volgt een analyse van het in de vorige paragraaf afgeleide aantal manieren waarop aan de n eindknopen van de a_n verschillende boomtopologieën twee verschillende klasselabels kunnen worden toegekend. Een restrictie daarbij is dat twee eindknopen met dezelfde beslisknoop als ouder, verschillende labels moeten dragen.

Analoog aan hetgeen in de vorige paragraaf op pagina 69 is afgeleid, is in te zien dat, wanneer dit aantal door b_n wordt genoteerd, geldt dat

$$b_n = \sum_{k=1}^{n-1} b_k b_{n-k}, \qquad n \geq 3, \tag{A.12}$$

waarbij voor de bomen T_1 en T_2 geldt dat $b_1 = b_2 = 2$. Wanneer voor de lege boom T_0 per definitie geldt dat $b_0 = 0$, dan kan vergelijking (A.12) worden geschreven als

$$b_n = \sum_{k=0}^{n} b_k b_{n-k}, \qquad n \geq 3. \tag{A.13}$$

De recurrente betrekking van vergelijking (A.13) kan met behulp van z-transformatie worden opgelost.

De unilateraal z-getransformeerde van de rij b_n is gedefinieerd als

$$B_I(z) := \sum_{n=0}^{\infty} b_n z^{-n}. \tag{A.14}$$

Substitutie van vergelijking (A.13) geeft

$$\begin{aligned} B_I(z) &= 2z^{-1} + 2z^{-2} + \sum_{n=3}^{\infty} \left(\sum_{k=0}^{n} b_k b_{n-k} \right) z^{-n} \\ &= 2z^{-1} - 2z^{-2} + \sum_{n=0}^{\infty} \left(\sum_{k=0}^{n} b_k b_{n-k} \right) z^{-n} \\ &= 2z^{-1}(1 - z^{-1}) + \left(\sum_{n=0}^{\infty} b_n z^{-n} \right)^2 \\ &= 2z^{-1}(1 - z^{-1}) + B_I^2(z). \end{aligned} \tag{A.15}$$

Oplossing van het tweedegraads polynoom in $B_I(z)$ van vergelijking (A.15) levert de wortels

$$B_I(z)_{1,2} = \frac{1}{2} \pm \frac{1}{2}\sqrt{1 - 8z^{-1}(1 - z^{-1})}. \tag{A.16}$$

Vanwege $b_0 = 0$ vervalt de oplossing met het + teken en is vergelijking (A.16) te schrijven als

$$B_I(z) = \frac{1}{2} - \frac{1}{2} W_I(z), \tag{A.17}$$

waarin $W_I(z) = (1 - 8z^{-1} + 8z^{-2})^{\frac{1}{2}}$.

De inverse z-getransformeerde van $W_I(z)$ kan op verschillende manieren worden bepaald:

Hypergeometrische functies

Het tweedegraads polynoom in z^{-1} onder het wortelteken van $W_I(z)$ is te ontbinden in factoren:

$$\begin{aligned} W_I(z) &= \sqrt{8z^{-2} - 8z^{-1} + 1} \\ &= \sqrt{8(z^{-1} - a)(z^{-1} - b)}, \end{aligned} \tag{A.18}$$

waarin $a = \frac{1}{2} - \frac{1}{4}\sqrt{2}$ en $b = \frac{1}{2} + \frac{1}{4}\sqrt{2}$. $W_I(z)$ is derhalve te schrijven als

$$W_I(z) = \left(1 - \frac{1}{a}z^{-1}\right)^{\frac{1}{2}} \left(1 - \frac{1}{b}z^{-1}\right)^{\frac{1}{2}}. \tag{A.19}$$

Uit vergelijking (A.6) en (A.14) volgt dat $W_I(z)$ het product is van twee z-getransformeerden. Volgens de convolutiestelling is de inverse z-getransformeerde van (A.19) gelijk aan

$$w_n = a_n * b_n, \tag{A.20}$$

waarin $a_n = \binom{\frac{1}{2}}{n}(-1)^n a^{-n}$ en $b_n = \binom{\frac{1}{2}}{n}(-1)^n b^{-n}$.

De convolutiesom van a_n en b_n is gedefinieerd als

$$a_n * b_n = \sum_{k=0}^{n} a_k b_{n-k}. \tag{A.21}$$

Substitutie van a_n en b_n in het rechterlid van (A.21) geeft

$$\begin{aligned} a_n * b_n &= \sum_{k=0}^{n} \binom{\frac{1}{2}}{k}(-1)^k a^{-k} \binom{\frac{1}{2}}{n-k}(-1)^{n-k} b^{k-n} \\ &= (-1)^n b^{-n} \sum_{k=0}^{n} \binom{\frac{1}{2}}{k} \binom{\frac{1}{2}}{n-k} \left(\frac{b}{a}\right)^k. \end{aligned} \tag{A.22}$$

Definieer nu $(c)_k := c(c+1)(c+2)\cdot \ldots \cdot(c+k-1)$. De hypergeometrische functie $F(a, b; c; z)$ kan dan in de vorm

$$F(a, b; c; z) = \sum_{n=0}^{\infty} \frac{(a)_n \cdot (b)_n}{n!(c)_n} z^n \tag{A.23}$$

worden genoteerd.

Vergelijking (A.22) is derhalve te schrijven als

$$\begin{aligned} a_n * b_n &= b^{-n} \sum_{k=0}^{n} \frac{(-\frac{1}{2})_k}{k!} \frac{(-\frac{1}{2})_{n-k}}{(n-k)!} \left(\frac{b}{a}\right)^k \\ &= \frac{(-\frac{1}{2})_n}{n!} b^{-n} \sum_{k=0}^{n} \frac{(-\frac{1}{2})_k (-n)_k}{k!(-n+\frac{3}{2})_k} \left(\frac{b}{a}\right)^k \\ &= \frac{(-\frac{1}{2})_n}{n!} b^{-n} F\left(-n, -\frac{1}{2}; -n+\frac{3}{2}; \frac{b}{a}\right) \qquad \text{(A.24)} \\ &= \frac{(-\frac{1}{2})_n}{n!} a^{-n} F\left(-n, -\frac{1}{2}; -n+\frac{3}{2}; \frac{a}{b}\right), \qquad \text{(A.25)} \end{aligned}$$

met $\frac{b}{a} = 3 + 2\sqrt{2}$, $\frac{a}{b} = 3 - 2\sqrt{2}$, $a^{-n} = 2^{3n/2}(1+\sqrt{2})^n$ en $b^{-n} = 2^{3n/2}(\sqrt{2}-1)^n$.

Gegenbauerpolynomen

Volgens [Magnus *et al.*, 1966, pagina 222] geldt dat

$$\begin{aligned} W_I(z) = (1 - 8z^{-1} + 8z^{-2})^{\frac{1}{2}} &= \left[1 - 2\sqrt{2}\cdot(2\sqrt{2}z^{-1}) + (2\sqrt{2}z^{-1})^2\right]^{\frac{1}{2}} \\ &= \sum_{n=0}^{\infty} C_n^{-\frac{1}{2}}(\sqrt{2})(2\sqrt{2}z^{-1})^n \\ &= \sum_{n=0}^{\infty} 2^{3n/2} C_n^{-\frac{1}{2}}(\sqrt{2})\, z^{-n}, \end{aligned} \tag{A.26}$$

waarin $C_n^{-\frac{1}{2}}(x)$ het Gegenbauerpolynoom van de graad n is.

Volgens (A.14) is het rechterlid van vergelijking (A.26) per definitie gelijk aan de unilateraal z-getransformeerde van $2^{3n/2}C_n^{-\frac{1}{2}}(\sqrt{2})$. Hieruit volgt door inverse z-transformatie van (A.17) dat

$$b_n = \frac{1}{2}\delta_n - 2^{3n/2}C_n^{-\frac{1}{2}}(\sqrt{2}), \tag{A.27}$$

waarin

$$\delta_n = \begin{cases} 1 & n = 0, \\ 0 & n \neq 0 \end{cases}.$$

Legendrepolynomen

Volgens [Magnus *et al.*, 1966, pagina 222] is het Gegenbauerpolynoom uit te drukken als een verschil van twee Legendrepolynomen:

$$\begin{aligned} C_n^{-\frac{1}{2}}(\sqrt{2}) &= \frac{-\frac{1}{2}}{n-\frac{1}{2}}\left[C_n^{\frac{1}{2}}(\sqrt{2}) - C_{n-2}^{\frac{1}{2}}(\sqrt{2})\right] \\ &= \frac{-1}{2n-1}\left[P_n(\sqrt{2}) - P_{n-2}(\sqrt{2})\right]. \end{aligned} \tag{A.28}$$

Hierin is $P_n(x)$ het Legendrepolynoom van de graad n. Vergelijking (A.27) is voor $n \geq 1$ op grond hiervan ook te schrijven als

$$b_n = \frac{2^{3n/2-1}}{2n-1}\left[P_n(\sqrt{2}) - P_{n-2}(\sqrt{2})\right]. \tag{A.29}$$

De oplossing in de vorm van Legendrepolynomen is de meest elegante.

Het aantal manieren waarop aan de a_n verschillende topologieën van binaire bomen met n beslisknopen twee klasselabels zijn toe te wijzen is derhalve gelijk aan

$$b_{n+1} = \frac{2^{3n/2+1}}{2n+1}\left[P_{n+1}(\sqrt{2}) - P_{n-1}(\sqrt{2})\right], \qquad n \geq 0. \tag{A.30}$$

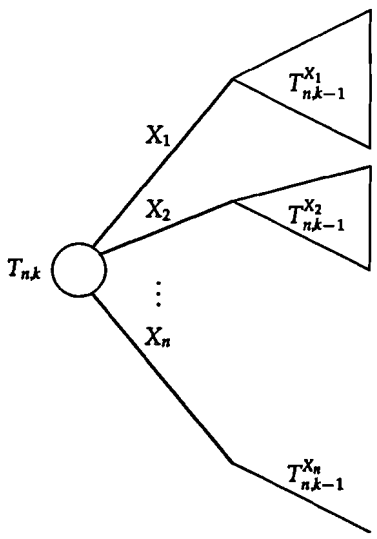
A.3 LOGISCH UNIEKE *K*-CONJUNCTIES

In deze paragraaf volgt een analyse van het aantal logisch unieke k-conjuncties van n kenmerken.

Veronderstel daartoe dat er n kenmerken $X_1, X_2, ..., X_n$ zijn waarvoor alle rijtjes van lengte k gegenereerd worden. Er zijn n^k van dergelijke rijtjes. Indien deze rijtjes als conjuncties van de kenmerken worden gezien, dan zijn er logisch gezien een aantal equivalent. Indien de equivalenten buiten beschouwing worden gelaten, worden de resterende conjuncties aangeduid als zijnde logisch uniek. De notatie voor het aantal logisch unieke k-conjuncties van n kenmerken is $c_{n,k}$. Wanneer verder het aantal logisch unieke k-conjuncties dat begint met het kenmerk X_i door $c_{n,k}^{X_i}$ wordt genoteerd, dan geldt

$$c_{n,k} = \sum_{i=1}^{n} c_{n,k}^{X_i}. \tag{A.31}$$

De $c_{n,k}$ unieke k-conjuncties worden voortgebracht door een boom $T_{n,k}$ van diepte k met n subbomen $T_{n,k-1}^{X_i}$ die op hun beurt alle logisch unieke rijtjes voortbrengen die beginnen met het kenmerk $X_i, X_{i+1}, ..., X_n$. Dit impliceert dat na een symbool X_i enkel en alleen de symbolen $X_i, X_{i+1}, ... X_n$ voorkomen. Een voorbeeld van een dergelijke boom is weergegeven in figuur A.2.



Figuur A.2 Een boom $T_{n,k}$ met subbomen $T_{n,k-1}^{X_i}$.

Voor $c_{n,k}^{X_i}$ geldt derhalve dat

$$c_{n,k}^{X_i} = \sum_{j=i}^{n} c_{n,k-1}^{X_j}, \qquad 1 \le i \le n, \quad k \ge 1. \tag{A.32}$$

Wanneer voor i in vergelijking (A.32) $n - i$ wordt gesubstitueerd en de eerste somterm wordt afgesplitst, is zij ook te schrijven als

$$\begin{aligned} c_{n,k}^{X_{n-i}} &= c_{n,k-1}^{X_{n-i}} + \sum_{j=n-i+1}^{n} c_{n,k-1}^{X_j}, \qquad 0 \le i < n, \quad k \ge 1 \\ &= c_{n,k-1}^{X_{n-i}} + c_{n,k}^{X_{n-i+1}}, \end{aligned} \tag{A.33}$$

waarbij $c_{n,0}^{X_j}$, $1 \le j \le n$, per definitie gelijk is aan 0. De recurrente betrekking van vergelijking (A.33) kan met behulp van z-transformatie worden opgelost.

De unilateraal z-getransformeerde van de rij $c_{n,k}^{X_{n-i}}$ is gedefinieerd als

$$C_I^{X_{n-i}}(z) := \sum_{k=0}^{\infty} c_{n,k}^{X_{n-i}} z^{-k}. \tag{A.34}$$

Substitutie van vergelijking (A.33) geeft

$$\begin{aligned} C_I^{X_{n-i}}(z) &= z^{-1} \sum_{k=0}^{\infty} c_{n,k}^{X_{n-i}} z^{-k} + \sum_{k=0}^{\infty} c_{n,k}^{X_{n-i+1}} z^{-k} \\ &= z^{-1} C_I^{X_{n-i}}(z) + C_I^{X_{n-i+1}}(z) \\ &= \frac{z}{z-1} C_I^{X_{n-i+1}}(z). \end{aligned} \tag{A.35}$$

Door inverse z-transformatie van vergelijking (A.35) volgt dat

$$\begin{aligned} c_{n,k}^{X_{n-i}} &= \sum_{j=0}^{k} c_{n,j}^{X_{n-i+1}} \\ &= \sum_{j=1}^{k} c_{n,j}^{X_{n-i+1}}. \end{aligned} \tag{A.36}$$

Indien $c_{n,k}^{X_{n-i}}$ met $d_i^{n,k}$ wordt aangeduid, dan is vergelijking (A.36) equivalent met

$$d_i^{n,k} = \sum_{j=1}^{k} d_{i-1}^{n,j}, \qquad i \ge 1, \quad k \ge 1, \tag{A.37}$$

waarbij $d_0^{n,j}$, $1 \le j \le k$, per definitie gelijk is aan 0. De recurrente betrekking van vergelijking (A.37) kan eveneens met behulp van z-transformatie worden opgelost.

De unilateraal z-getransformeerde van de rij $d_i^{n,k}$ is gedefinieerd als

$$D_I^{n,k}(z) := \sum_{i=0}^{\infty} d_i^{n,k} z^{-k}. \tag{A.38}$$

Substitutie van vergelijking (A.37) geeft

$$D_I^{n,k}(z) = 1 + \sum_{i=1}^{\infty} \left(\sum_{j=1}^{k} d_{i-1}^{n,j} \right) z^{-i}$$

$$
\begin{aligned}
&= 1+\sum_{j=1}^{k}\left(\sum_{i=1}^{\infty} d_{i-1}^{n,j} z^{-i}\right)\\
&= 1+\sum_{j=1}^{k}\left(z^{-1}\sum_{i=0}^{\infty} d_{i}^{n,j} z^{-i}\right)\\
&= 1+z^{-1}\sum_{j=1}^{k} D_I^{n,j}(z)\\
&= 1+z^{-1}\left(\sum_{j=1}^{k-1} D_I^{n,j}(z)+D_I^{n,k}(z)\right)\\
&= \frac{z}{z-1}+\frac{1}{z-1}\sum_{j=1}^{k-1} D_I^{n,j}(z). \qquad \text{(A.39)}
\end{aligned}
$$

Successieve substitutie van $D_I^{n,k-1}(z), D_I^{n,k-2}(z), \ldots, D_I^{n,1}(z)$ in vergelijking (A.39) geeft

$$D_I^{n,k}(z)=\frac{z}{z-1}+\sum_{j=1}^{k-2}\frac{z^j}{(z-1)^{j+1}}+\frac{z^{k-2}}{(z-1)^{k-1}}D_I^{n,1}(z), \qquad \text{(A.40)}$$

waarbij $D_I^{n,1}(z)=\sum_{i=0}^{\infty} d_i^{n,1} z^{-i}$ en $d_i^{n,1}=d_{i-1}^{n,1}=\ldots=d_0^{n,1}=1$, zodat $D_I^{n,1}(z)=\frac{z}{z-1}$.

Vergelijking (A.40) is derhalve te schrijven als

$$D_I^{n,k}(z)=\frac{z}{z-1}+\sum_{j=1}^{k-1}\frac{z^j}{(z-1)^{j+1}}. \qquad \text{(A.41)}$$

Door inverse z-transformatie van vergelijking (A.41) volgt dat

$$
\begin{aligned}
d_i^{n,k} &= 1+\sum_{j=1}^{k-1}\binom{i+j-1}{j}\\
&= \sum_{j=0}^{k-1}\binom{i+j-1}{j} \qquad \text{(A.42)}\\
&= \sum_{j=0}^{k-1}\binom{i+j-1}{i-1}\\
&= 1+\sum_{j=0}^{k-2}\binom{i+j}{i-1}. \qquad \text{(A.43)}
\end{aligned}
$$

Volgens de driehoek van Pascal geldt voor gehele p en q, $p \geq q$, dat

$$\binom{p}{q}+\binom{p}{q-1}=\binom{p+1}{q}. \qquad \text{(A.44)}$$

Indien hierin p door $p+1$ wordt vervangen, dan volgt dat

$$\binom{p+1}{q}+\binom{p+1}{q-1}=\binom{p+2}{q}. \qquad \text{(A.45)}$$

Substitutie van (A.44) in vergelijking (A.45) levert

$$\binom{p}{q-1} + \binom{p+1}{q-1} = \binom{p+2}{q} - \binom{p}{q}.$$

Algemeen geldt dat

$$\sum_{j=0}^{r} \binom{p+j}{q-1} = \binom{p+r+1}{q} - \binom{p}{q}. \tag{A.46}$$

Door substitutie van (A.46) met $p = q = i$ en $r = k-2$ in vergelijking (A.43) volgt dat

$$d_i^{n,k} = \binom{k+i-1}{i}. \tag{A.47}$$

Op grond van de identiteit $c_{n,k}^{X_{n-i}} = d_i^{n,k}$ geldt dus ook dat

$$c_{n,k}^{X_{n-i}} = \binom{k+i-1}{i}, \qquad 0 \le i < n. \tag{A.48}$$

Substitutie van (A.48) in vergelijking (A.31) levert het aantal logisch unieke k-conjuncties van n testvariabelen

$$\begin{aligned} c_{n,k} &= \sum_{i=1}^{n} c_{n,k}^{X_i} \\ &= \sum_{i=0}^{n-1} c_{n,k}^{X_{n-i}} \\ &= \sum_{i=0}^{n-1} \binom{k+i-1}{i}. \end{aligned}$$

Analoog aan (A.42) en (A.47) volgt hieruit dat

$$c_{n,k} = \binom{k+n-1}{k}. \tag{A.49}$$

BIJLAGE B

Geselecteerde gegevensbestanden

B.1 INLEIDING

In deze bijlage volgt een overzicht van de geselecteerde gegevensbestanden, waarvan in de experimenten gebruik is gemaakt. In een eerste paragraaf worden de specificaties van de bestanden beschreven, waarna in een tweede paragraaf een overzicht van eerder in de literatuur gepubliceerde classificatieresultaten volgt.

B.2 SPECIFICATIES

In het hiernavolgende worden de karakteristieken van de gegevensbestanden beknopt beschreven aan de hand van een aantal kenmerkende grootheden. Bovendien is voor elk bestand een bronvermelding weergegeven. Waar mogelijk zijn referenties naar eerder gerapporteerde classificatieresultaten vermeld.

B.2.1 BU: Bupa leverstoornissen

Bron: Bupa Medical Research Ltd.
R.S. Forsyth
8 Grosvenor Avenue
Mapperley Park
Nottingham NG3 5DX
Mei, 1990.

Eerder gebruik: Anders dan in PC/BEAGLE User's Guide (R. Forsyth) onbekend

Aantal objecten: 345

Aantal kenmerken: 7 (inclusief klasselabel)

Kenmerkinformatie:

1	MCV	mean corpuscular volume
2	ALKPHOS	alkaline phosphotase
3	SGPT	alamine aminotransferase
4	SGOT	asparate aminitransferase
5	GAMMAGT	gamma-glutamyl transpeptidase
6	DRINKS	number of half-pint equivalents of alcoholic beverages drunk per day
7	SELECTOR	class label

Missing values: geen

B.2.2 GL2: Glass identificatie

Bron: Central Research Establishment
Home Office Forensic Science Service
Aldermaston
Reading
Berkshire RG7 4PN
September, 1987

Eerder gebruik: o.a. [Evett & Spiehler, 1987], [Holte, 1993]

Aantal objecten: 214

Aantal kenmerken: 10 (inclusief klasselabel)

Kenmerkinformatie:

1	ID	1-214
2	RI	refractive index
3	NA	Sodium
4	MG	Magnesium
5	AL	Aluminium
6	SI	Silicon
7	K	Potassium
8	CA	Calcium
9	BA	Barium
10	FE	Iron
11	CLASS	1 building windows float processed
		2 building windows non float processed
		3 vehicle windows float processed
		4 vehicle windows non float processed
		5 containers
		6 tableware
		7 headlamps

Missing values: geen

Opmerkingen: In het gebruikte GL2-bestand zijn de klassen 1 en 3 gecombineerd en zijn de klassen 4 tot en met 7 verwijderd, zoals in [Holte, 1993].

B.2.3 HE: Hepatitis domein

Bron: G. Gong, Carnegie-Mellon University, via
B. Cestnik
Jozef Stefan Institute
Jamova 39
61000 Ljubljana
Yugoslavia
November, 1988.

Eerder gebruik: o.a. [Diaconis & Efron, 1983] en [Cestnik *et al.*, 1987]

Aantal objecten: 155

Aantal kenmerken: 20 (inclusief klasselabel)

Kenmerkinformatie:

1	CLASS	DIE, LIVE
2	AGE	10, 20, 30, 40, 50, 60, 70, 80
3	SEX	male, female
4	STEROID	no, yes
5	ANTIVIRALS	no, yes
6	FATIGUE	no, yes
7	MALAISE	no, yes
8	ANOREXIA	no, yes
9	LIVER BIG	no, yes
10	LIVER FIRM	no, yes
11	SPLEEN PALPABLE	no, yes
12	SPIDERS	no, yes
13	ASCITES	no, yes
14	VARICES	no, yes
15	BILIRUBIN	0.39, 0.80, 1.20, 2.00, 3.00, 4.00
16	ALK PHOSPHATE	33, 80, 120, 160, 200, 250
17	SGOT	13, 100, 200, 300, 400, 500
18	ALBUMIN	2.1, 3.0, 3.8, 4.5, 5.0, 6.0
19	PROTIME	10, 20, 30, 40, 50, 60, 70, 80, 90
20	HISTOLOGY	no, yes

Missing values: 75

Klasseverdeling: `DIE`: 32, `LIVE`: 123

B.2.4 IR: Iris domein

Bron: R.A. Fisher, 1936.

Eerder gebruik: o.a. [Fisher, 1936] en [Duda & Hart, 1973]

Aantal objecten: 150

Aantal kenmerken: 5 (inclusief klasselabel)

Kenmerkinformatie:

1	SEPAL LENGTH	sepal length (cm)
2	SEPAL WIDTH	sepal width (cm)
3	PETAL LENGTH	petal length (cm)
4	PETAL WIDTH	petal width (cm)
5	CLASS	1 Iris Setosa
		2 Iris Versicolor
		3 Iris Virginica

Missing values: geen

Klasseverdeling: 33.3% voor elk van de drie klassen

B.2.5 PI: Pima Indians Diabetes domein

Bron: National Institute of Diabetes and Digestive and Kidney Diseases
V. Sigillitto
Applied Physics Laboratory
The John Hopkins University
John Hopkins Road
Laurel, MD 20707
Mei, 1990.

Eerder gebruik: [Smith *et al.*, 1988]

Aantal objecten: 768

Aantal kenmerken: 9 (inclusief klasselabel)

Kenmerkinformatie:

1	NUM PREGNANT	number of times pregnant
2	GLUCOSE	plasma glucose concentration
3	PRESSURE	diastolic blood pressure (mm Hg)
4	TRICEPS	triceps skin fold thickness (mm)
5	INSULIN	2-hour serum insulin (mu U/ml)
6	MASS INDEX	body mass index (weight in kg/(height in m)^ 2)
7	PEDIGREE	diabetes pedigree function
8	AGE	age (years)
9	CLASS	0 tested negative for Diabetes Mellitus
		1 tested positive for Diabetes Mellitus

Missing values: geen

Klasseverdeling: 0: 500, 1: 268

B.3 OVERZICHT VAN EERDER GEPUBLICEERDE RESULTATEN

In het navolgende wordt voor de geselecteerde bestanden die in de experimenten zijn gebruikt, een beknopt overzicht gepresenteerd van de classificatieresultaten die eerder in de literatuur zijn gerapporteerd. Opgemerkt dient te worden dat bij het construeren van de classificatoren als optimaliseringscriterium uitsluitend maximalisatie van de classificatienauwkeurigheid is gebruikt.

De in tabel B.1 weergegeven schattingen van de nauwkeurigheid zijn verkregen door een *random subsampling* experiment 25 maal te herhalen en de resultaten daarvan te middelen. Alle gegevens in de tabel zijn ontleend aan [Holte, 1993].

Tabel B.1 Classificatieresultaten van uit de literatuur bekende algoritmen.

Bestand	Nauwkeurigheid	Algoritme	Referentie
BU	–	–	–
GL2	0.743	C4 (post-pruning)	[Holte, 1993]
	0.770	1R*	[Holte, 1993]
	0.785	1RW	[Holte, 1993]
HE	0.801	CN2 (ongeordend, laplace)	[Clark & Boswell, 1991]
	0.812	C4 (post-pruning)	[Holte, 1993]
	0.830	ASSISTANT (pre-pruning)	[Cestnik *et al.*, 1987]
	0.845	1RW	[Holte, 1993]
	0.851	1R*	[Holte, 1993]
IR	0.938	C4 (post-pruning)	[Holte, 1993]
	0.940	ID3 (pruned)	[Buntine, 1989]
	0.953	CART (cross-validation)	[Weiss & Kapouleas, 1989]
	0.959	1R*	[Holte, 1993]
	0.960	PVM	[Weiss *et al.*, 1987]
	0.960	1RW	[Holte, 1993]
PI	0.760	ADAP	[Smith *et al.*, 1988]

BIJLAGE C

Programmatuur

C.1 INLEIDING

EXPLORE omvat een aantal programma's voor het inductief leren van logische beslisregels. Deze bijlage bevat een beschrijving van de diverse bestanden die worden gebruikt en geeft een beknopt overzicht van de programmacode die ten behoeve van dit onderzoek is ontwikkeld.

C.2 BESCHRIJVING VAN BESTANDEN

In het hiernavolgende wordt een functionele beschrijving van de executeerbare bestanden gegeven. De in- en uitvoerbestanden die door EXPLORE worden gebruikt, komen daarna aan de orde.

C.2.1 Executeerbare bestanden

De executeerbare bestanden worden onderverdeeld in twee categorieën: programma- en scriptbestanden. De verzameling programmabestanden bestaat uit routines voor het construeren en evalueren van beslisregels, alsmede voor het partitioneren van een gegevensbestand van voorbeeldobjecten. De serie scriptbestanden omvat een routine voor het uitvoeren van de werkwijze voor het bepalen van de optimale beslisregel (zie pagina 40) en een routine voor het presenteren van de resultaten daarvan.

Programmabestanden

Explore

Synopsis: `explore` [*opties*] `-l`<*argument*> <*bestandsnaam*>

Beschrijving: Heuristische zoekprocedure voor het op basis van gegevensbestand <*bestandsnaam*> construeren van logische beslisregels van lengte <*argument*>.

Opties:

-t{*cd*}	regels in CNF- (*c*) of DNF-vorm (*d*); bij verstek worden regels in zowel CNF- als DNF-vorm gegenereerd.
-o{*mwd*}	kies één rekenkundige operator op basis van de Mediaan- (*m*), *W*- (*w*) of Diagonaal-methode (*d*).
-n*i*	minimum vereist aantal objecten *i* tussen twee opeenvolgende beslisdrempels; verstekwaarde = 1.

Invoer:

- gegevensbestand '<*bestandsnaam*>`.data`'
- specificatiebestand '<*bestandsnaam*>`.spec`'

Uitvoer:

- lijstbestand '<*bestandsnaam*>`.list`'
- regelbestand '<*bestandsnaam*>`.rule`'

Resample

Synopsis: `resample` <*bestandsnaam*> <*argument 1*> <*argument 2*>

Beschrijving: Routine voor het steekproefsgewijs, gestratificeerd in <*argument 1*> partities indelen van de voorbeeldobjecten uit gegevensbestand <*bestandsnaam*>. De steekproef is aselect; de toevalsgenerator wordt met <*argument 2*> gestart.

Invoer:

- gegevensbestand '<*bestandsnaam*>`.data`'

Uitvoer:

- partitioneringsbestand '<*bestandsnaam*>`.part`'

Tryall

Synopsis: `tryall -l`*<argument>* *<bestandsnaam>*

Beschrijving: Uitputtende zoekprocedure voor het op basis van gegevensbestand *<bestandsnaam>* construeren van alle mogelijke logische beslisregels van lengte *<argument>*.

Invoer:

- gegevensbestand '*<bestandsnaam>*`.data`'
- specificatiebestand '*<bestandsnaam>*`.spec`'

Uitvoer:

- lijstbestand '*<bestandsnaam>*`.list`'
- regelbestand '*<bestandsnaam>*`.rule`'

Validate

Synopsis: `validate -l`*<argument>* *<bestandsnaam>*

Beschrijving: Routine voor het op basis van gegevensbestand *<bestandsnaam>* bepalen van de prestaties van de beste regel van lengte *<argument>*.

Invoer:

- regelbestand '*<bestandsnaam>*`.rule`'
- testbestand '*<bestandsnaam>*`.data.tst`'

Uitvoer:

- prestatiemaat

Scriptbestanden

Extractrules.pl

Synopsis: `extractrules.pl` <*bestandsnaam*> <*argument 1*> <*argument 2*>

Beschrijving: Routine voor het extraheren van de beste beslisregel uit de door <*argument 2*>-voudige *cross-validation*, tot en met regellengte <*argument 1*> voortgebrachte lijstbestanden.

Invoer:

- lijstbestand '<*bestandsnaam*>`.list.1.1`'

  ⋮
- lijstbestand '<*bestandsnaam*>`.list.`<*argument 1*>`.`<*argument 2*>'

Uitvoer:

- lijstbestand '<*bestandsnaam*>`.rules.1`'

  ⋮
- lijstbestand '<*bestandsnaam*>`.rules.`<*argument 1*>'

Rulefit_cv.pl

Synopsis: `rulefit_cv.pl` <*bestandsnaam*> <*argument 1*> <*argument 2*>

Beschrijving: Routine voor het op basis van gegevensbestand <*bestandsnaam*> door <*argument 1*>-voudige *cross-validation* bepaalde optimale regellengte, *k*, construeren van een optimale beslisregel. De toevalsgenerator voor het partitioneren van het gegevensbestand wordt met <*argument 2*> gestart.

Invoer:

- gegevensbestand '<*bestandsnaam*>`.data`'
- specificatiebestand '<*bestandsnaam*>`.spec`'

Uitvoer:

- partitioneringsbestand '<*bestandsnaam*>`.part`'
- lijstbestand '<*bestandsnaam*>`.list.1.1`'

  ⋮
- lijstbestand '<*bestandsnaam*>`.list.`<*k*>`.`<*argument 2*>'
- prestatiemaatbestand '<*bestandsnaam*>`.out`'

C.2.2 Invoerbestanden

Als invoerbestanden maakt EXPLORE gebruik van een gegevensbestand dat voorbeeldobjecten bevat en een specificatiebestand dat, naast informatie over het gegevensbestand, de prestatiemaatrestricties bevat.

Gegevensbestand

Het gegevensbestand bevat de voorbeeldobjecten waarvan in de analyse gebruik wordt gemaakt. De gegevens dienen in matrixvorm te worden gerepresenteerd, waarbij de objecten in de rijen, en de kenmerken in de kolommen zijn geplaatst. Kenmerkwaarden zijn continu en/of binair, doch niet nominaal. *Missing values* worden niet toegestaan.

Specificatiebestand

Het specificatiebestand bevat een beschrijving van het te analyseren gegevensbestand en een specificatie van de prestatiemaatrestricties. Trefwoorden geven het aantal kenmerken, de kenmerknamen, de naam van het gegevensbestand en de restricties aan.

De restricties zijn de minimum vereiste waarden voor sensitiviteit (SE), specificiteit (SP), positief predictieve waarde (PPV), negatief predictieve waarde (NPV) en nauwkeurigheid (ACC).

Als voorbeeld is in figuur C.1 het specificatiebestand dat voor experiment #5 (zie pagina 55) is gebruikt, weergegeven.

```
FEATURES = 8
    NUM_PREGNANT
    GLUCOSE
    PRESSURE
    TRICEPS
    INSULIN
    MASS_INDEX
    PEDIGREE
    AGE
DATAFILE = ../data/pi.data

SP  >= 0.000
PPV >= 0.000
NPV >= 0.000
ACC >= 0.743
```

Figuur C.1 Voorbeeld van een specificatiebestand.

C.2.3 Uitvoerbestanden

De reeks uitvoerbestanden die EXPLORE voortbrengt, bestaat uit een lijstbestand van veelbelovende beslisregels, een regelbestand dat de beste regel uit het lijstbestand bevat, een partitioneringsbestand dat aangeeft hoe de voorbeeldobjecten in een leer- en testbestand worden gepartitioneerd, en een prestatiemaatbestand dat *cross-validation* schattingen van de te optimaliseren prestatiemaat bevat.

Lijstbestand

Het lijstbestand bevat voor elk kenmerk: het aantal potentiële drempelwaarden, een schatting van de oppervlakte onder de ROC-curve, een indicatie of de ROC-curve geheel boven of onder de diagonaal ligt, en de rekenkundige operatoren die worden gebruikt. Daarnaast geeft het bestand het aantal potentiële beslisregels aan. Als laatste bevat het een lijst van veelbelovende beslisregels en hun respectieve prestatiematen: sensitiviteit (SE), specificiteit (SP), positief predictieve waarde (PPV), negatief predictieve waarde (NPV) en nauwkeurigheid (ACC).

Regelbestand

Het regelbestand bevat de beste beslisregel uit het lijstbestand in een voor de machine te lezen representatie. Dit bestand wordt gebruikt om de prestaties van de regel op het leerbestand te bepalen.

Partitioneringsbestand

Het partitioneringsbestand bevat indices die aangeven hoe de voorbeeldobjecten uit het gegevensbestand bij *cross-validation* worden gepartitioneerd.

Prestatiemaatbestand

Het prestatiemaatbestand bevat de resultaten van een door *cross-validation* bepaalde optimale beslisregel. Voor elke testpartitie is een schatting van de te optimaliseren prestatiemaat aangegeven.

C.3 INVENTARIS VAN PROGRAMMACODE

Tabel C.1 en C.2 geven respectievelijk een beknopt overzicht van de bron- en scriptbestanden die in C en PERL zijn gecodeerd.

Tabel C.1 Inventaris van bronbestanden in C.

Naam	Omschrijving
`cmplx.c`	Functie voor berekening complexiteit probleemdomein
`combin.c`	Functies voor combinatorische routines
`distrib.c`	Functies voor berekening parameters univariate verdeling
`error.c`	Functie voor afhandeling van fouten
`explore.c`	Programma voor heuristisch genereren van beslisregels
`finput.c`	Functies voor invoer van gegevensbestand
`resample.c`	Programma voor steekproefsgewijs partitioneren van gegevensbestand
`roc.c`	Functies voor berekening parameters van ROC-curve
`rva.c`	Functie voor berekening van potentiële drempelwaarden
`set.c`	Functies voor bewerkingen op verzamelingen
`tryall.c`	Programma voor uitputtend genereert van beslisregels
`validate.c`	Programma voor valideren van geïnduceerde beslisregel

Tabel C.2 Inventaris van scriptbestanden in PERL.

Naam	Beschrijving
`extractrules.pl`	Routine voor extraheren regel uit lijstbestanden
`rulefit_cv.pl`	Routine voor *cross-validation* experimenten