Katholieke Hogeschool Sint-Lieven
Departement Industrieel Ingenieur
Opleiding Elektronica-ICT
Afstudeerrichting informatie- en communicatietechnieken
Gebroeders Desmetstraat 1, 9000 Gent

Genereren van Medische Kennis uit Praktijkdata

Eindverhandeling ingediend tot het behalen van
de graad van Master in Industriële Wetenschappen:
Elektronica-ICT
en aangeboden door: Boris De Vloed

Promotoren: dr. ir. Annemie Vorstermans
ir. Jos De Roo
Copromotor: lic. ing. Joris Maervoet

Academiejaar 2008 - 2009

Dankwoord

Vooreerst wil ik mijn externe promotor ir. Jos De Roo (Agfa HealthCare) bedanken zonder wie dit eindwerk onmogelijk zou zijn. Ik dank hem voor het delen van zijn kennis, het geven van tips en advies doorheen het volledige eindwerk. Deze inbreng was van essentieel belang.

Verder wil ik ook nog dr. ir. Annemie Vorstermans (KaHo Sint-Lieven) bedanken voor het van nabij opvolgen van de thesis en de zeer gewaardeerde feedback op eerdere versies van deze scriptie. Dit heeft bijgedragen tot de kwaliteit en de leesbaarheid van de scriptie.

Ook lic. ing. Joris Maervoet (KaHo Sint-Lieven) ben ik dankbaar om mij wegwijs te maken in het logisch programmeren en het geven van feedback.

Een vierde woord van dank gaat naar dr. Hans Cools (Agfa HealthCare) voor het valideren van de gevonden resultaten en het geven van concrete tips.

Abstract

In hospitalen worden heel wat gegevens bijgehouden over diagnoses, behandelingen en resultaten van patiënten. Door het toepassen van dataminingtechnieken kan kennis uit deze praktijkdata worden gehaald. Het doel van dit eindwerk is om in opdracht van Agfa HealthCare een model op te stellen om de relaties tussen bepaalde aandoeningen en symptomen voor te stellen. In dit eindwerk werd gekozen voor Bayesaanse netwerken. Deze kunnen dan worden getransformeerd en gebruikt om artsen te adviseren bij diagnosestelling en behandelingen.

Er zal getracht worden deze modellen te verifiëren door deze te vergelijken met de resultaten uit bestaande data mining tools. De universele bruikbaarheid van het model kan getest worden door modellen uit verschillende databanken met elkaar te vergelijken. De validatie van de resultaten zal gebeuren door deze voor te leggen aan specialisten binnen het medische domein.

Daarna zal onderzocht worden hoe deze statische Bayesaanse netwerken dynamisch gemaakt kunnen worden. Dit is aangewezen omdat bij vele aandoeningen de symptomen veranderen in de tijd. Een veelbelovende mogelijkheid om dit te realiseren is het gebruik van abductive logic programming. Dit zal in deze thesis onderzocht worden.

Trefwoorden: Datamining, Bayesaanse Netwerken, Medische Kennis, Abductive Logic Programming

Inhoudsopgave

Lijst van figuren

Lijst van tabellen

Lijst van gebruikte afkortingen

ALP	Abductive Logic Programming
CDR	Clinical Data Depository
CIS	Clinical Information System
CRISP-DM	Cross-Industry Standard Process for Data Mining
FCM	Fuzzy Cognitive Maps
FN	False Negatives
FP	False Positives
ILP	Inductive Logic Programming
MCC	Matthews Correlation Coefficient
MDL	Minimum Description Length
N3	Notation 3
RDF	Resource Description Framework
TDHD	Top Directed Hypothesis Derivation
TAN	Tree Augmented Naive Bayes
TN	True Negatives
TP	True Positives

Hoofdstuk 1

Inleiding

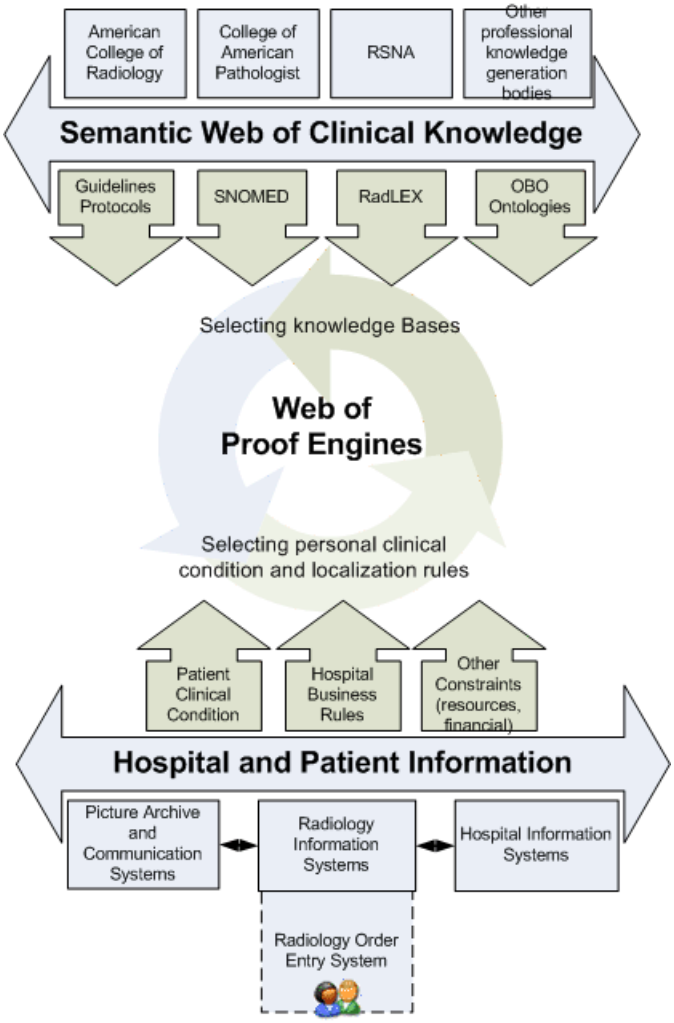
1.1 Situering

Dit eindwerk kadert binnen een breder decision support system voor artsen (Chen and De Roo [7]) zoals bijv. ORBIS©Medical en wordt geschreven in opdracht van Agfa HealthCare. Het descission support system geeft onder andere aan hoeveel zekerder men wordt van een bepaalde diagnose bij het al dan niet uitvoeren van bijkomende tests en wat de kosten ervan zijn. Deze informatie helpt artsen om een diagnose te stellen of om te beslissen verdere tests te laten uitvoeren. Dit systeem wordt gevoed door klinische kennis afkomstig van klinische richtlijnen en verwijzingscriteria (referral criteria). Deze data worden manueel door artsen verzameld en omgezet naar machine readable data. In figuur 1.1 is een voorbeeld van zo'n systeem, specifiek voor radiologie weergegeven. Deze werkwijze heeft enkele nadelen. Vooreerst is dit zeer arbeidsintensief. Daarenboven lopen deze richtlijnen soms achter op nieuwe protocollen en mogelijkheden, zoals bvb. nieuwe beeldvormingsmogelijkheden in de radiologie.

Vanuit deze situatie ontstond de wens om aanvullend ook medische kennis rechtstreeks te kunnen genereren uit gegevens over symptomen en vastgestelde aandoeningen bij patiënten. Deze data zullen typisch uit een *Clinical Data Repository (*CDR*)* afkomstig zijn. In het DEBUGIT-project (Detecting and Eliminating Bacteria UsinG Information Technology [18]) is deze Clinical Data Repository een geünificeerde voorstelling van alle klinische data uit de heterogene verzameling van verschillende *Clinical Information Systems (*CIS*)* van de deelnemende partners. Het was initieel bij dit eindwerk ook de bedoeling om databanken afkomstig uit hospitalen van het DEBUGIT-project te gebruiken. Echter, door het uitblijven van bruikbare data is er uiteindelijk gekozen voor een veel beperktere publieke databank.

Het descission support system van Agfa HealthCare (Chen and De Roo [7]) gebruikt een *semantic reasoner* over verschillende kennisbanken. Deze kennisbanken zijn opgebouwd uit regels in N3 (Berners-Lee and Connolly [4]) die een *belief network* voorstellen. Belief networks of Bayesaanse netwerken worden in latere hoofdstukken nog uitvoerig besproken. Concreet bevat de kennisbank regels zoals deze:

```
({{:Fever :assessedBy :Exam} e:boolean e:T} {:Flu e:boolean e:T}) e:conditional 0.99.
({{:Fever :assessedBy :Exam} e:boolean e:T} {:Flu e:boolean e:F}) e:conditional 0.5.
```



Figuur 1.1: Een voorbeeld van een Decision Support System (uit [7])

De eerste regel beschrijft de kans dat griep koorts veroorzaakt tewijl de tweede regel beschrijft wat de kans is dat koorts wordt veroorzaakt door alles behalve griep. Een bespreking van de N3 syntax vindt u in bijlage.

1.2 Opdrachtomschrijving

De bedoeling van dit eindwerk is om na te gaan hoe door datamingtechnieken toe te passen op klinische data een additionele kennisbron (knowledgebase) gegenereerd kan worden. Het resultaat van de datamining zijn verbanden tussen aandoeningen en observaties (zoals geslacht, leeftijd, symptomen, resultaten van tests enz.). Deze kennis moet in de vorm van *Notation3* (zie bijlage) regels worden opgeslagen zodat het door een semantische reasoner

gelezen en begrepen kan worden. Hoewel het uiteindelijk de bedoeling is om kennis te gebruiken voor decision support, waar het belangrijk is te weten welke observaties op welke aandoeningen wijzen, zal de kennisbron bijhouden welke observaties gepaard gaan met welke aandoeningen. Zo blijft contextonafhankelijkheid en stabiliteit van de gevonden kennis gegarandeerd. De instabiliteit van kennis die vertrekt van de observaties wordt duidelijk in het geval van griep. Tijdens een griepepidemie zal een heel groot percentage van de patiënten die een arts consulteren als symptoom onder andere koorts hebben en zal griep worden gediagnostiseerd. Wanneer er geen griepepidemie is, is de kans dat koorts een andere oorzaak heeft dan weer veel groter. Al die tijd is de kans dat griep koorts veroorzaakt echter (zo goed als) gelijk gebleven.

De gegenereerde kennis zal in een zodanige vorm moeten zijn dat deze niet alleen de werkelijkheid accuraat modelleert zodat er uiteindelijk goede decision support geleverd kan worden. Omdat het decision support systeem in real time adviezen berekent, zal de kennis in een zodanig model vervat moeten zijn dat er geen combinatorische explosie optreedt wanneer de kennisbron zeer veel regels bevat. Zowel de schaalbaarheid als de precisie zullen moeten worden afgewogen.

Om dit concept uit te werken zal datamining toepast worden op een databank met gegevens over hartfalen. Er zal getracht worden complexere regels waarbij rekening gehouden wordt met onderlinge afhankelijkheid tussen observaties, te vinden. Zo kan er bijvoorbeeld een afhankelijkheid gevonden worden tussen leeftijd en cholestorolgehalte. Regels die met meerdere observaties rekening houden zijn preciezer en daarom ook waardevoller bij decision support.

De resultaten zullen gevalideerd worden door specialisten uit het medische domein.

1.3 Overzicht

Deze scriptie bestaat uit twee experimenten. In een eerste experiment wordt een zelfgeschreven programma gebruikt om naïeve Bayesaanse netwerken te genereren uit medische data.

In een tweede experiment wordt voortgebouwd op de ervaring uit het eerst experiment. Op zoek naar resultaten die de resultaten naïeve Bayesaanse netwerken overtreffen wordt gebruik gemaakt van premissen om specifiekere naïeve Bayesaanse netwerken te genereren. Zo kan b.v.b. een apart Bayesaans netwerk opgesteld worden per geslacht door in de premisse na te gaan of de patiënt een man of een vrouw is. Om die premissen te vinden die de beste resultaten opleveren zal de mogelijkheid van logisch programmeren nagegaan worden.

In het volgende hoofdstuk wordt eerst de gebruikte dataset besproken. Zowel de betekenis van de data, de kwaliteit ervan en enkele eerste inzichten uit de histogrammen worden besproken.

Hoofdstuk 2

Data preparation en -understanding

2.1 Inleiding

Als dataset werd gekozen voor de *Heart Disease Data Set* van de *UCI Machine Learning Repository* [2]. Dit is een verzameling van vier datasets die voor dit project samengevoegd werden tot één dataset. De originele CSV-bestanden zijn beschikbaar op de UCI-repository, alsook op de bijgevoegde CD-ROM waar ook een SQL-dump van de gebruikte databank is toegevoegd.

In totaal bevat deze dataset 920 entries en origineel 14 kolommen. De betekenis en de verschillende mogelijke waarden per kolom worden verduidelijkt in tabel 2.1.

2.1.1 Kwaliteit van de Data

In de gegevens zitten enkele fouten. In de Zwitsere databank zijn de ingevulde cholesterolgehalten gelijk aan 0, wat onmogelijk is, en in de Long Beach databank is er een patiënt waarbij de bloeddruk tijdens rust gelijk is aan 0 mm/Hg. Deze anomalieën werden verwijderd. Deze velden worden nu als missing values gezien.

Niet alle records zijn volledig ingevuld. In tabel 2.2 wordt per kolom van iedere dataset weergegeven hoeveel waarden er ook werkelijk ingevuld zijn. Daaruit blijkt dat enkel de Cleveland dataset zo goed als volledig ingevuld is. Vooral de velden die de helling van het ST-segment (slope) (zie bijlage 2) en het aantal gekleurde bloedvaten (ca) weergeven, vertonen heel wat hiaten bij de andere datasets. In de Zwitserse dataset ontbreken ook heel wat suikerspiegels (fbs), en zoals reeds eerder vermeld, alle cholesterolwaarden.

2.1.2 Grafische weergave van de dataset

Er werden histogrammen opgesteld die de verdeling weegeven per parameter van de patiënten met en zonder hartfalen. Deze voorstelling laat ons toe om de correlatie tussen de verschillende parameters en het hartfalen in te schatten. Een volledig overzicht van de histogrammen vindt u in de bijlage. Figuur 2.1 geeft al enkele van de belangrijke

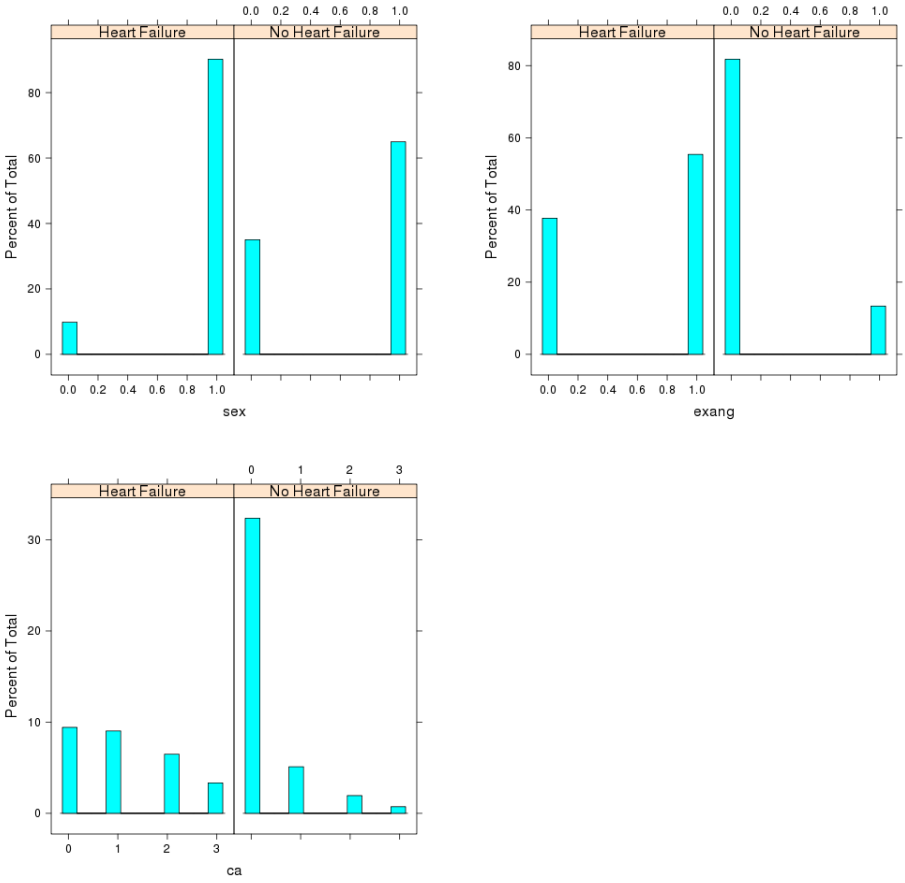
KOLOMNAAM	DATATYPE	BETEKENIS
patientid [1]	numeriek	Sleutel om iedere rij uniek te identificeren
source [2]	[1 - 4]	1 = Cleveland Clinic Foundation 2 = Hungarian Institute of Cardiology. Budapest 3 = University Hospitals, Zurich and Basel 4 = V.A. Medical Center, Long Beach
age	numeriek	leeftijd
sex	binair	0 = vrouwelijk, 1 = mannelijk
cp (chestpain)	[1 - 4]	1 = typical angina 2 = atypical angina 3 = non-anginal pain 4 = asymptomatic
trestbps	numeriek	bloeddruk tijdens rust bij opname in het ziekenhuis [mm Hg]
chol	numeriek	cholesterolgehalte in serum [mg/dl]
fbs (fasting blood sugar)	binair	nuchtere bloedglucosespiegel $<$120 mg/dl 0 = false, 1 = true
restecg	[0 - 2]	ECG tijdens rust 0 = normaal 1 = ST-T golf abnormaliteit 2 = waarschijnlijke of zekere hypertrofie
thalach	numeriek	maximaal behaalde hartslag
exang	binair	door inspanning uitgelokte angina
oldpeak	numeriek	relatieve daling van het ST-segment tegenover rust (zie bijlage 2)
slope	[1 - 3]	helling van het ST-segment bij inspanning (zie bijlage 2) 1 = up 2 = flat 3 = down
ca	[0 - 3]	aantal door fluoroscopie gekleurde belangrijke bloedvaten
thal		3 = normal 6 = fixed defect 7 = reversable defect
num	numeriek	diagnose van hartfalen (angiographic disease status) +1 per bloedvaten dat $>$50% vernauwd is.

[1] deze sleutel werd zelf toegevoegd.
[2] Deze kolom is zelf toegevoegd na het samenvoegen van de vier datatsets tot een groter geheel.

Tabel 2.1: Beschrijving van de Heart Disease Data Set

	PROCESSED.CLEVELAND.DATA		PROCESSED.HUNGARIAN.DATA		PROCESSED.SWITZERLAND.DATA		PROCESSED.VA.DATA		TOTAAL	
	abs [303]	rel [%]	abs [294]	rel [%]	abs [123]	rel [%]	abs [200]	rel [%]	abs [920]	rel [%]
age	303	100	294	100	123	100	200	100	920	100
sex	303	100	294	100	123	100	200	100	920	100
cp	303	100	294	100	123	100	200	100	920	100
trestbps	303	100	293	99,7	121	98,4	143	71,5	860	93,5
chol	303	100	271	92,2	0	0	193	96,5	718	78,0
fbs	303	100	286	97,3	48	39	193	96,5	830	90,2
restecg	303	100	293	99,7	122	99,2	200	100	918	99,8
thalach	303	100	293	99,7	122	99,2	147	73,5	865	94,0
exang	303	100	293	99,7	122	99,2	147	73,5	865	94,0
oldpeak	303	100	294	100	117	95,1	144	72	858	93,3
slope	303	100	104	35,4	106	86,2	98	49	611	66,4
ca	299	98,7	3	1,0	5	4,07	2	1	309	33,6
num	303	100	294	100	123	100	200	100	920	100

Tabel 2.2: Volledigheid van de Heart Disease Data Set



Figuur 2.1: Enkele historgrammen

histogrammen weer. Men kan nu al zien dat er bijvoorbeeld een positieve correlatie is met hartfalen bij mannen en een negatieve correlatie voor patiënten die geen geïnduceerde angina of geen door fluoroscopie gekleurde belangrijke bloedvaten hebben.

2.1.3 Voorbereiding van de data

De *Heart Disease Data Set* van de *UCI Machine Learning Repository* [2] bevat enkel ruwe data in CSV formaat.

De vier CSV-bestanden worden aan elkaar geplakt, nadat eerst voor iedere lijn van de oorspronkelijke files een nummer werd geplakt waardoor de bron van iedere entry te achterhalen blijft.

Het verkregen CSV-bestand werd geïmporteerd in MySQL. Tot slot werden de anomalieën vervangen door NULL met standaard SQL UPDATE-query's.

Hoofdstuk 3

Generatie van algemene kennis

3.1 Doelstelling

In eerste instantie is het de bedoeling logische regels te genereren die het verband tussen aandoeningen en observaties weergeven. Zoals reeds eerder vermeld in de inleiding moet de kennis stabiel zijn, m.a.w. gaan over de observatie die bij een aandoening hoort. Omdat de kennis gevalideerd moet kunnen worden, is het nodig om transparante, expliciete kennis te hebben. Zogenaamde "black box"modellen zijn dus niet gewenst. Een derde eis is dat de kennis gebruikt kan worden als kennisbron voor de semantische reasoner. De kennis zal dus als Notation3 regels geëxporteerd moeten kunnen worden. De laatste en belangrijkste beperking is de combinatorische complexiteit bij het omzetten van de kennis van stabiel $aandoening \rightarrow observatie$ kennis naar voor diagnostiek bruikbare $observatie \rightarrow aandoening$ kennis. Deze omzetting gebeurt in real time en het resultaat moet voor de gebruiker onmiddellijk beschikbaar zijn.

De gebruikte dataset is de publiek beschikbare *Heart Disease Data Set* van de *UCI Machine Learning Repository* (Aha [2]).

3.2 Technologieën

3.2.1 Inleiding

Zoals reeds in de inleiding aangehaald is het de bedoeling dat patiëntengegevens uit een databank omgezet worden naar medische kennis. Er werd geen open-source programma gevonden waarvan de licentie het toeliet het programma aan te passen en te gebruiken binnen een gedeeltelijke closed source omgeving. Daarom werd er gekozen om zelf een programma te schrijven in de plaats van een bestaand programma zoals Weka uit te breiden.

In de modelleerfase worden meestal een of meerdere kennismodellen en dataminingtechnieken onderzocht waarna de meestbelovende oplossingen geselecteerd worden. Hier is de keuzevrijheid voor het kennismodel van in het begin van het project al

beperkt. Vermits de semantische reasoner Bayesaanse regels verwacht, zullen we dus een soort Bayesaans netwerk moeten kiezen. Toch zullen in dit hoofdstuk kort enkele andere kennismodellen besproken worden.

Voor een groter scala aan technieken en grondigere studie hiervan wordt verwezen naar [26], [15] en [16].

3.2.2 Decision Trees

Een decision tree is een boomstructuur waar in ieder knooppunt een attribuut geëvalueerd wordt. Meestal houdt dit in dat een attribuut met een constante vergeleken wordt, het is evenwel ook mogelijk dat verschillende attributen met elkaar vergeleken worden of er een functie gebruikt wordt die verschillende attributen combineert. De bladeren van decision tree zijn klassen waar iedere instantie die dit knooppunt bereikt toe behoort [26]. Een voorbeeld van een decision tree is het Goldman Chest Pain Protocol in figuur 3.1. Door in ieder knooppunt de conditie na te gaan en aan de hand daarvan een tak te kiezen, komt men uiteindelijk bij een diagnose. Deze vorm van redeneren is in tegenspraak met een van de doelstellingen, namelijk het genereren van stabiele kennis. Naargelang de context zal een andere boom nodig zijn, b.v.b. bij het al dan niet heersen van een griepepidemie.

Enkele veelgebruikte algoritmen om deze bomen op te stellen zijn ID3, C4.5 en C5.0 [26]. Een meer diepgaande bespreking hiervan valt buiten de scope van deze tekst.

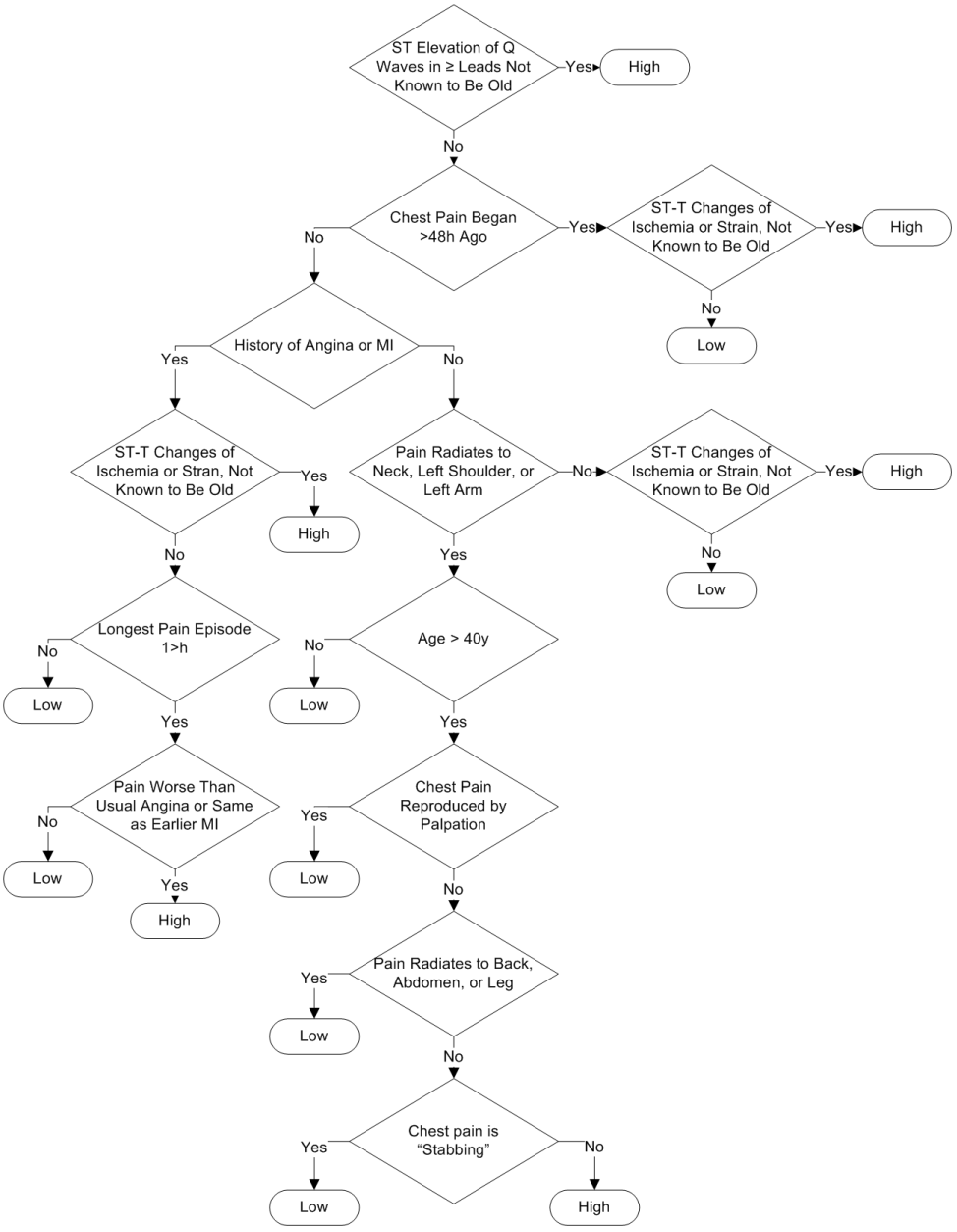
3.2.3 Neurale Netwerken

Er zijn heel veel verschillende soorten neurale netwerken. Dit eindwerk beperkt zich tot het algemeen principe en waarom er uiteindelijk gekozen is om niet met neurale netwerken verder te gaan. Een diepgaandere bespreking is te vinden in [15] en [26]. Een neuron in een artificieel neuraal netwerk krijgt inputs van een stroomopwaarts gelegen neuron, of rechtstreeks uit een dataset. De inputs worden gecombineerd en gebruikt als invoer voor een activatiefunctie (meestal een sigmoide of een stapfunctie). Dit resulteert in een output die dan wordt doorgegeven als input naar stroomafwaarts gelegen neuronen. Aan iedere verbinding hangt een gewicht. Het neurale netwerk leert door de output van voorbeelden te vergelijken met de gewenste output en de gewichten in functie hiervan aan te passen.

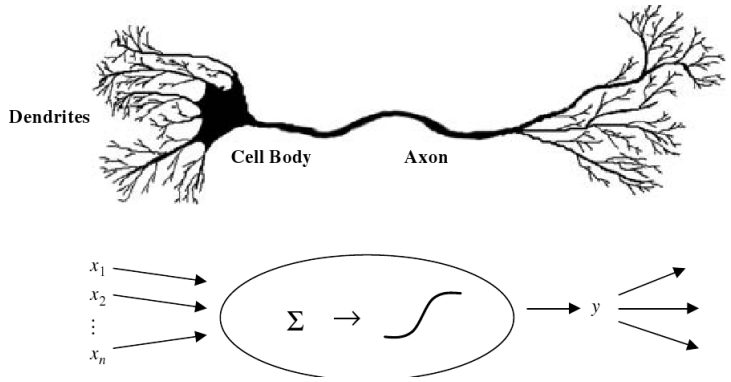
Neurale netwerken geven in tegenstelling tot decision trees of Bayesaanse regels geen transparante kennis. Omdat de kennis in de vorm van aan de verbindingen toegekende gewichten moeilijk te vergelijken is met kennis uit klinische trials, is het zo goed als onmogelijke om neurale netwerken op een voldoende wijze te valideren. Daarom zullen zogenaamde “black box ”systemen zoals neurale netwerken, maar ook support vector machines [26] niet verder onderzocht worden.

3.2.4 Bayesaanse Netwerken

Bayesaanse netwerken zijn gerichte grafen waar de edges probabilistische relaties tussen de nodes weergeven. Figuur 3.4 op pagina 13 is een eenvoudig voorbeeld van hoe zo’n



Figuur 3.1: Het Goldman Chest Pain Protocol uit Cannon [6]



Figuur 3.2: Een artificieel neuron (uit [15])

	Actual == True	Actual == False
Predicted == True	True Positives	False Positives
Predicted == False	False Negatives	True Negatives

Tabel 3.1: Confusion Matrix

netwerk er uit kan zien.

Ieder knooppunt van waaruit een edge vertrekt, kan gezien worden als een test. De uitkomst daarvan kan correct of fout zijn. Dit wordt samengevat in tabel 3.1.

Aan de edges worden zowel een *sensitiviteit* als een *aspecificiteit* toegekend.

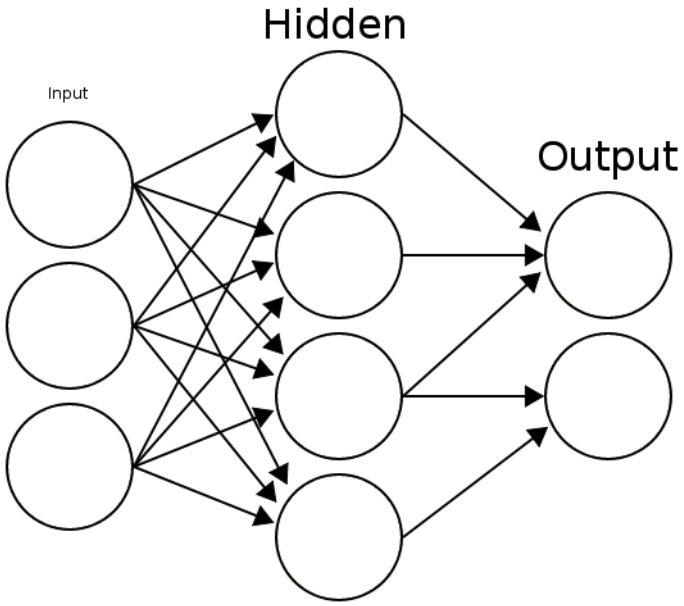
sensitiviteit (s) De kans dat een aandoening een bepaald gevolg veroorzaakt.

$$s = P(test = true|actual = true) = \frac{TP}{TP+FN} \tag{3.1}$$

aspecificiteit (a) De kans dat alles behalve een bepaalde aandoening een bepaald gevolg veroorzaakt.

$$a = 1 - P(test = false|actual = false) = 1 - \frac{TN}{TN+FP} \tag{3.2}$$

De sensitiviteit en aspecificiteit kunnen geplot worden op een *receiver operating characteristic* (ROC curve). De stippellijn op de figuur, waar de sensitiviteit gelijk is aan de aspecificiteit is de is de *no-discrimination line*. Een test op die lijn heeft dezelfde voorspellende waarde als een compleet random event, zoals b.v.b. het werpen van een muntstuk. De linkerbovenhoek, waar de sensitiviteit 1 is en de aspecificiteit 0 is het meest "ideale"punt. In dat geval zijn er geen vals positieven of negatieven. Hoe dichter een test bij dit ideale punt ligt, hoe groter de voorspellende waarde ervan.



Figuur 3.3: Een artificieel neuraal netwerk

Theorema van Bayes

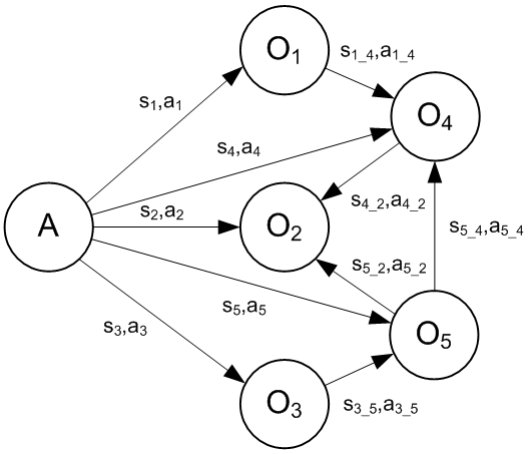
Zeer belangrijk voor dit eindwerk is het feit dat de regel van Bayes gebruikt kan worden om kennis over observaties, gegeven een aandoening omgezet kan worden naar de kans op een aandoening, gegeven bepaalde observaties.

Het theorema van Bayes volgt rechtstreeks uit de definitie van de voorwaardelijke kans [11].

$$P(A \cap B) = P(A|B) \cdot P(B) \tag{3.3}$$
$$= P(B|A) \cdot P(A) \tag{3.4}$$
$$P(A|B) = \frac{P(B|A) \cdot P(A)}{P(B)} \tag{3.5}$$

Concreet voor observatie O, aandoening A, sensitiviteit s en aspecificiteit a.

$$P(A|O) = \frac{P(O|A) \cdot P(A)}{P(O)} \tag{3.6}$$
$$= \frac{P(O|A) \cdot P(A)}{P(O|A) \cdot P(A) + P(O|\overline{A}) \cdot P(\overline{A})} \tag{3.7}$$
$$= \frac{P(O|A) \cdot P(A)}{P(O|A) \cdot P(A) + P(O|\overline{A}) \cdot (1 - P(A))} \tag{3.8}$$
$$= \frac{s \cdot P(A)}{s \cdot P(A) + a \cdot (1 - P(A))} \tag{3.9}$$
$$\tag{3.10}$$



Figuur 3.4: Een eenvoudig Bayesaans netwerk

$P(A)$, de kans op aandoening A kan op het moment zelf in rekening gebracht worden. Om bij het griepvoorbeeld te blijven. Tijdens een griepepidemie zal $P(A)$ vrij hoog zijn, maar op een ander moment veel kleiner.

Algoritmen

Een complete graaf is meestal niet wenselijk. Om het aantal verbindingen tussen de knooppunten te beperken tot die verbindingen die waardevolle informatie bevatten, kan het K2 algoritme gebruikt worden [26]. Het K2 algoritme laat toe om een Bayesaans netwerk te vinden met een minimale entropie. Door het verminderen van het aantal verbindingen vermindert ook het aantal waarden die geëvalueerd worden bij het doorlopen van de graaf.

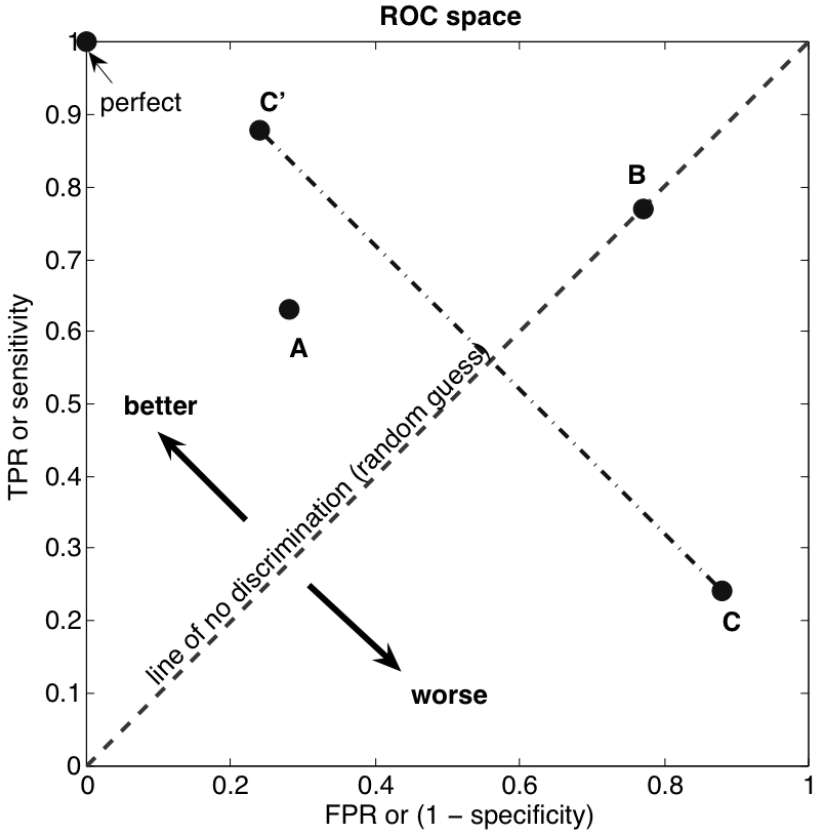
Het feit dat alle nodes verschillende edges kunnen hebben, leidt tot een combinatorische explosie. Het berekenen van dit netwerk zelf stelde voor de gebruikte dataset geen enkel probleem. Maar omdat in de praktijk gebleken is dat redeneren over zulke netwerken ondoenbaar is, zal een ander soort Bayesaans netwerk gekozen moeten worden.

Naïeve Bayesaanse Netwerken

Het andere uiterste is wanneer verondersteld wordt dat alle oorzaken onafhankelijk zijn van elkaar. Wanneer O_i en O_j twee verschillende observaties zijn bij een bepaalde aandoening A. Dan zijn O_i en O_j onafhankelijk van elkaar wanneer geldt: $P(O_i|A) = P(O_i|O_j,A)$ wanneer $P(O_i|O_j) > 0$ voor alle mogelijke waarden van O_i, O_j en A. Met andere woorden, wanneer de kans dat observaties O_i wordt waargenomen onafhankelijk is van het feit dat O_j waargenomen werd.

De aanname van conditionele onafhankelijkheid zal meestal niet kloppen. Veelal zal een van deze drie soorten conditionele afhankelijkheden optreden:

equivalentie wanneer een concept vertegenwoordigd wordt door meerdere variabelen.



Figuur 3.5: ROC-curve

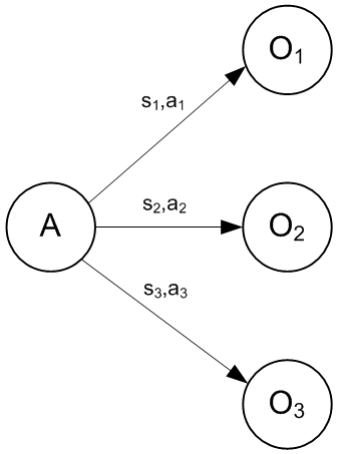
bvb.: lichaamstemperatuur $>37°$C en koorts

subklasse wanneer een variabele afhankelijk is van een andere variabele.
bvb.: clusterhoofdpijn is een subklasse van hoofdpijn

complement wanneer (Boolse) variabelen elkaar uitsluiten.
bvb.: man en vrouw

Het weglaten van alle edges tussen de observaties, m.a.w. het negeren van onderlinge afhankelijkheid brengt informatieverlies met zich mee. Zo kan er bijvoorbeeld een verband zijn tussen leeftijd en bloeddruk. Toch doet deze verregaande vereenvoudiging het goed in tal van applicaties. Zo kunnen naïeve Bayesaanse classifiers nog altijd concurreren met complexere classifiers zoals C4.5 (Friedman [9] en Rish [22]). Een klassiek voorbeeld waar naïeve Bayesaanse netwerken gebruikt worden, zijn spamfilters zoals o.a. SpamAssasin en Bogofilter [1].

De schaalbaarheid volgt direct uit de structuur van naïeve Bayes :$O(n)$, het aantal edges is beperkt tot 1 verbinding tussen iedere observatie en de aandoening. Dit in tegenstelling tot het andere extreme, een compleet Bayesaans netwerk waar de complexiteit $O(n!)$ is.



Figuur 3.6: Een eenvoudig naïef Bayesaans netwerk

3.3 Implementatie

3.3.1 Inleiding

Om de hierboven vernoemde doelen te bereiken zal een Java programma geschreven worden dat via JDBC een verbinding maakt met een databank. De keuze voor Java brengt enkele voordelen met zich mee. Een eerste zeer groot voordeel is de platformonafhankelijkheid. Een ander belangrijk punt is het voorhanden zijn van enkele zeer degelijke en uitgebreide gratis ontwikkelingsomgevingen voor deze taal. Als databanktechnologie werd initieel gekozen voor het filegebaseerde *SQLite* omwille van de eenvoud en het feit dat er zeer weinig administratie aan is. Echter, naarmate het programma complexer werd, bleek dat een klassieke client-server databank veel schaalbaarder was en minder geheugenproblemen gaf dan de JDBC/SQLITE-combinatie. Vermits de databank eenvoudig is en er geen objecten moeten worden in opgeslagen en dergelijke meer, is het voor deze thesis niet nodig om een ORM-raamwerk zoals Hibernate te gebruiken. In de plaats daarvan werd een lichte `DatabaseConnection`-laag geschreven die minder impact heeft op de performantie.

Binarisatie van continue grootheden

Vele kolommen zoals leeftijd en cholesterolgehalte bevatten continue waarden. Om de sensitiviteit en aspecificiteit tussen de verschillende nodes te berekenen moeten deze bereiken vertaald worden naar Boolse expressies die waar of vals kunnen zijn. Zo zullen de cholesterolgehalten worden opgesplitst in een risicogroep en een niet risicogroep. Om dit

domein op te splitsen werd bij Agfa HealthCare de volgende heuristiek gebruikt:

```
x is aantal patiënten met een aandoening;
y is totaal aantal patiënten;
Sorteer patiënten volgens parameter;
if positieve correlatie parameter en aandoening then
    cut-off is waarde van de parameter bij patiënt nummer x ;
else
    cut-off is waarde van de parameter bij patiënt nummer (y-x) ;
```

Algoritme 1: Heuristiek voor binarisatie

Deze heuristiek gaf meestal vrij goede resultaten, voor sommige kolommen waren de resultaten dezelfde als die van een state of the art tool zoals Weka. Meestal waren de resultaten echter in de buurt van wat Weka als cut-off gebruikte, maar de resultaten van de classificatie waren meestal iets slechter. Daarom werd in het Java-programma dat in het kader van dit eindwerk werd geschreven deze heuristiek gebruikt om een goede beginwaarde te vinden vanwaar met een hillclimbing na enkele iteraties resultaten gevonden werden die vergelijkbaar zijn met die van Weka. In figuur 3.7 op de volgende pagina is dit grafisch weergegeven.

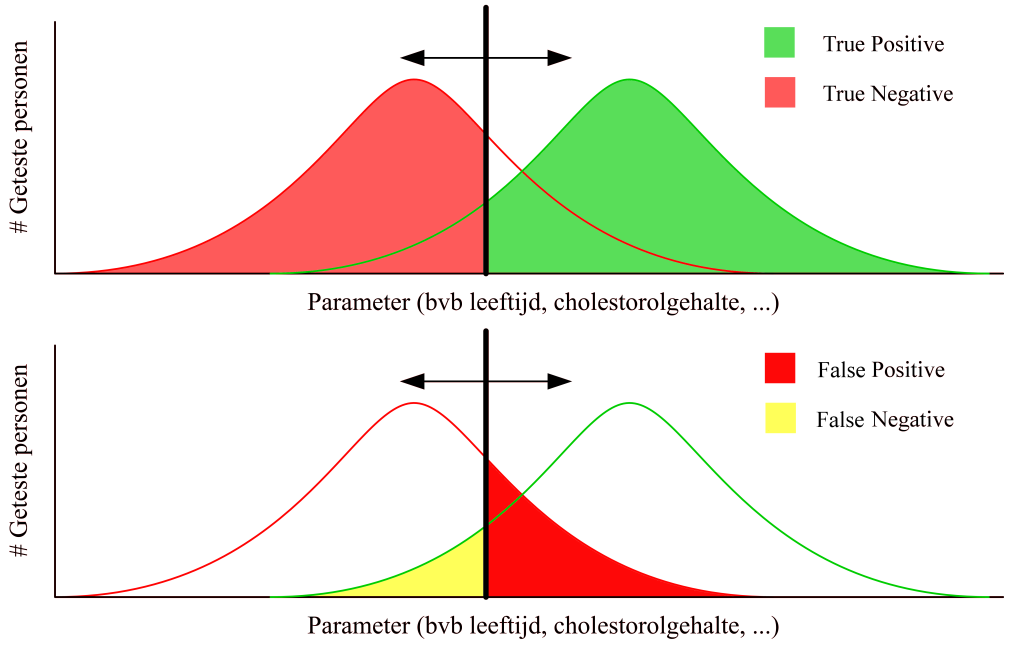
Als objectieffunctie werd de *Matthews Correlation Coefficient (*MCC*)* gebruikt, de MCC reduceert de volledige confusion matrix tot 1 parameter. Deze is gelijk aan nul wanneer er geen correlatie is, gelijk aan 1 bij een perfecte correlatie en gelijk aan -1 wanneer er een perfect inverse correlatie is. In het geval dat 1 van de sommen in de noemer gelijk is aan nul, wordt de noemer arbitrair gelijkgesteld aan 1 waardoor de MCC nul wordt. Dit is het geval omdat de som enkel gelijk kan zijn aan nul, wanneer beide termen gelijk zijn aan nul vermits deze aantallen geen negatieve waarden kunnen aannemen.

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}} \tag{3.11}$$

Opsplitsen van grootheden in verschillende klassen Niet alle kolommen met numerieke waarden stellen ook numerieke parameters voor. Zoals uit tabel 3.1 blijkt, slaan de verschillende waarden van `cp` op verschillende locaties waar de patiënt pijn voelt na een zware fysieke inspanning. Het gebruiken van een cut-off waarde om zulke kolommen te binariseren zou betekenisloze “kennis”genereren. Zulke kolommen worden opgesplitst in verschillende Boolse variabelen, zo zal de `cp`-kolom worden vertaald naar 4 verschillende soorten pijn die elk aan -of afwezig kunnen zijn.

3.3.2 Architectuur

In de figuur 3.8 is een overzicht gegeven van de verschillende klassen van het geschreven programma. Omwille van de leesbaarheid zijn de attributen en methoden weggelaten. Op de cd-rom is een gedetailleerdere klassendiagram toegevoegd.



Figuur 3.7: Binaire Classificatie

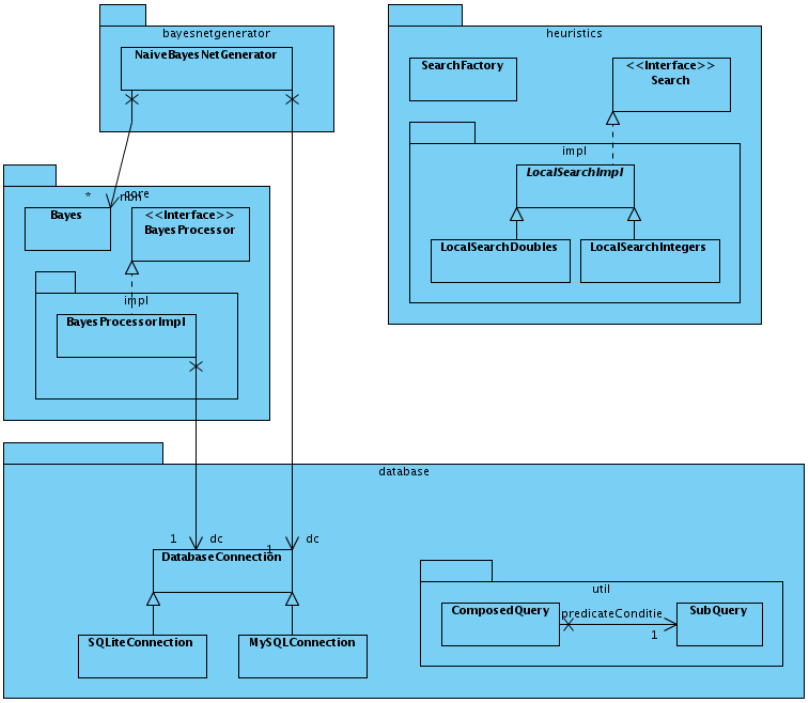
NaiveBayesNetGenerator

De `NaiveBayesNetGenerator` vertaalt de vrij abstracte input van de gebruiker naar methodeaanroepen van `BayesProcessor`. De invoer van de gebruiker wordt in meer detail besproken in de sectie *Gebruik*. `NaiveBayesNetGenerator` houdt ook het gevonden netwerk bij als een `List` van `Bayes`- elementen en bevat de nodige methoden om de gevonden kennis af te drukken, onder andere als N3.

Bayes en BayesProcessor

`Bayes`-objecten bevatten de eigenlijke kennis over de correlatie tussen de aandoening en een observatie. De objecten bevatten vanzelfsprekend de naam van de aandoening, de observatie, de sensitiviteit, aspecificiteit, maar ook hoe de observatie opgedeeld is in klassen en de confusion-matrix. `Bayes`-objecten bevatten naast de constructoren, getters en setters enkel methoden om zichzelf af te drukken.

De logica werd verschoven naar de klasse `BayesProcessor`. De klasse `BayesProcessor` bevat de meest essentiële operaties, zoals het binariseren van data of opsplitsen in meerdere klassen en het berekenen van de confusion-matrix.



Figuur 3.8: Klassendiagram (overzicht)

LocalSearch

Om de binarisatie van de observatieparameters te optimaliseren werd gebruik gemaakt van het hillclimbing zoals besproken in het hoofstuk over modellering. De methode `findMax(Bayes bayes, DatabaseConnection dc)` is dezelfde voor `Doubles` en `Integers` en wordt geïmplementeerd door de klasse `LocalSearchImpl`. De implementatie van de methoden om de neighbourhood te vergelijken verschilt voor `Doubles` en `Integers` en wordt geïmplementeerd door de klassen `LocalSearchDoubles` en `LocalSearchIntegers`.

`LocalSearchImpl` implementeert de interface `Search` en de juiste `Search`-klasse wordt teruggegeven via de `SearchFactory`. Zo kan indien later blijkt wanneer hillclimbing niet voldoet, makkelijk een andere heuristiek of een exhaustieve brute force berekening voor de splitwaarde worden ingeplugd.

DatabaseConnection

Als extra laag tussen de eigenlijke applicatie en de JDBC-verbinding met de databank werd de `DatabaseConnection` klasse ingevoerd. Hierdoor kunnen de SQL-statements en de Java-code gescheiden blijven. Dit zorgt voor meer transparantie en minder kans op (typ)fouten. Ook de omslachtige JDBC-specifieke commando's zoals `ResultSets` die telkens gesloten moeten worden, worden zo uit de rest van het programma gehouden.

`DatabaseConnection` bevat ook een Bools attribuut `verbose` waarmee alle query's en hun resultaten weggegeschreven worden naar de console. Dit om het debuggen van semantische fouten te vergemakkelijken.

Er zijn momenteel twee klassen die overerven van `DatabaseConnection`, nl.: `MySQLConnection` en `SQLiteConnection` enkel de constructor hiervan verschilt, zo moet er onder andere een gebruikersnaam en een paswoord worden meegegeven bij een MySQL verbinding en niet bij een SQLite verbinding.

ComposedQuery en SubQuery

`ComposedQuery` en `SubQuery` werden in het leven geroepen om te vermijden dat in de code stukken SQL zoals `NOT(statement)` of `statement1 AND statement2` zouden sluipen. Bij het concateneren van zulke stukken SQL wordt al gauw een spatie vergeten waardoor er SQL-syntax errors optreden.

Een `SubQuery` bevat slechts twee attributen, een `String statement` en een `boolean` die aangeeft of al dan niet de negatie van dit statement bedoeld is. `ComposedQuery` erft over van `ArrayList<SubQuery>` en laat toe een predicaat mee te geven (dat `null` mag zijn) als `String` in de constructor. De methode `toString()` van `ComposedQuery` print een correct SQL-statement met de inhoud van de `ArrayList` en het predicaat. Bij het wissen van een `ComposedQuery` via de `clear()`-methode worden alle `SubQuery`'s verwijderd, maar blijft het predicaat behouden. Dit is handig om moeilijk opspoorbare semantische fouten te vermijden wanneer een predicaat werd vergeten meegeven in een query.

3.3.3 Gebruik

Om een naïef Bayesaans netwerk te genereren hoeft de gebruiker enkel de volgende parameters mee te geven aan de constructor van de `NaiveBayesNetGenerator`:

- een `DatabaseConnection`-object
- De Boolse expressie die aangeeft dat de aandoening aanwezig is, in deze dataset is dat `num>0`, m.a.w. minstens één bloedvat met meer dan 50% vernauwing.
- Indien er een of meerdere kolommen waarden bevatten die niet als getallen geïnterpreteerd kunnen worden, moeten die ook worden meegegeven. bvb de `cp` kolom waar de verschillende getallen op verschillende soorten pijn slaan.

Optioneel kunnen ook de volgende parameters meegegeven worden:

- De niet relevante kolommen, bvb `source` en `patientid` zijn geen medische observaties.
- Een premisse, zo is het mogelijk om een specifiek model te berekenen dat enkel geldt voor mannen door `sex=1` mee te geven.

De methode `calculateNaiveBayesianNetwork` geeft een `ArrayList` terug met `Bayes`-objecten.

Het is ook mogelijk om de methode `getParameters` van de klasse `BayesProcessor` rechtstreeks aan te spreken. Zo kunnen de sensitiviteit en aspecificiteit voor andere dan de optimale opsplitsing ook berekend worden. Een motivatie hiervoor kan zijn het gebruik van een vooraf gedefinieerde klasse. Zo is het denkbaar dat men niet geïnterreseerd is voor de sensitiviteit en aspecificiteit voor de volgens de MCC optimale opsplitsing, maar bijvoorbeeld data over de leeftijdscategorie ouder dan 60 wenst.

3.4 Evaluatie

De correctheid van de gevonden waarden uit de databank werd binnen het programma zelf op twee manieren nagegaan. Met `setVerbose(true)` werden de gebruikte query's en de resultaten ervan afgedrukt op het scherm. Dit liet toe om op een transparante manier de correctheid van de gebruikte query's na te gaan. JUnit werd gebruikt om op een eenvoudige manier enkele tests te doen, b.v.b. het nagaan van de juiste intiële threshold, of het feit dat TP + TN + FP + FN == N waar is N het totaal aantal niet NULL entries voor de observatie in kwestie is.

Bij het berekenen van de sensitiviteiten werd ervoor gekozen om de observaties `cp`, `restecg`, `slope` en `thal` niet te laten opsplitsen in twee klassen, maar voor iedere mogelijke waarde van de observatie een aparte klasse te gebruiken. De noodzaak hiervan voor `cp` volgde duidelijk uit de beschrijving van de dataset. Voor de andere observaties werd het advies van een arts gevolgd. De resultaten hiervan zijn te vinden in tabel 3.4.

Om de gevonden thresholds bij de binarisatie te verifiëren werd de `bayes.BayesNet`-classifiers van Weka [25] gebruikt (Bouckaert [5]). Hoewel de classifier van Weka naar een model *observatie* → *aandoening* bouwt in tegenstelling tot deze thesis waar een model *aandoening* → *observatie* gegenereerd wordt, wordt toch eenzelfde opdeling in klassen verwacht. De MCC-score functie die gebruikt werd voor deze thesis is onafhankelijk van de zin van de edges in de graaf. Het veranderen van de zin van de edge in een Bayesaans netwerk komt overeen met het verwisselen van `actual` en `predicted` in de confusion matrix (tabel 3.1), of nog het verwisselen van FP en FN in de Matthews Correlation Coefficient (3.11). Men kan nu makkelijk inzien dat de waarde van de MCC gelijk blijft in beide gevallen.

Weka vindt dezelfde threshold-waarde voor `age`. De parameters `trestbps` en `chol` werden niet in klassen opgedeeld, waardoor er geen vergelijking gemaakt kan worden. Er werden verschillende score functies en instelling van de classifier geprobeerd zoals beschreven in [5], zonder veel resultaat. Dit valt te verklaren door de lage informatiewaarde van deze edges, dit komt ook tot uiting in de lage MCC waarden voor deze parameters.

Bij een evaluatie van de stabiliteit van de threshold waarden door verschillende subsets van de dataset te vergelijken, bleken gevonden thresholds vrij stabiel zoals te zien is in tabel 3.3. Dit ondanks de beperkte hoeveelheid data. Zo werd in één subset de threshold leeftijd 56 gevonden i.p.v. 55 en in een andere een threshold voor het cholesterolgehalte 246 i.p.v.

CONDITIE	SENSITIVITEIT	ASPECIFICITEIT	MCC
age >= 55	0.7	0.42	0.29
sex = 1	0.64	0.26	0.31
cp = 1	0.44	0.56	-0.06
cp = 2	0.14	0.66	-0.41
cp = 3	0.36	0.61	-0.21
cp = 4	0.8	0.28	0.52
trestbps >= 132	0.62	0.5	0.12
chol >= 241	0.55	0.43	0.13
fbs = 1	0.69	0.49	0.15
restecg = 0	0.52	0.62	-0.1
restecg = 1	0.66	0.53	0.11
restecg = 2	0.57	0.55	0.02
thalach <142	0.71	0.37	0.35
exang = 1	0.84	0.37	0.47
oldpeak >= 0.889	0.77	0.37	0.4
slope = 1	0.39	0.78	-0.39
slope = 2	0.78	0.48	0.31
slope = 3	0.78	0.63	0.1
ca >= 1	0.75	0.27	0.48
thal = 3	0.3	0.8	-0.51
thal = 6	0.77	0.55	0.14
thal = 7	0.81	0.39	0.42

Tabel 3.2: Resultaten van naïeve Bayes

parameter	VOLLEDIGE DATASET	SUBSET 1	SUBSET 2
age	55	55	56
trestbps	132	134	125
chol	241	241	246
thalach	142	140	142
oldpeak	0.889	0.600	0.933

Tabel 3.3: Stabiliteit van de thresholds

241.

De sensitiviteiten en de aspecificiteiten in de subsets bleken ook stabiel te zijn zolang er genoeg data voorhanden waren. Kolommen met te veel `null`-waarden en/of met kleine klassen (zoals b.v.b. `cp`) vertonen grotere schommelingen. Het valt te verwachten dat deze fenomenen weg zullen ebben wanneer er met grotere en meer realistische datasets gewerkt wordt. Een volledig overzicht van de resultaten is bijgevoegd op de cd-rom.

Validatie

Bij de validatie van de resultaten door een arts van Agfa HealthCare werden geen discrepanties tussen de data en de resultaten gevonden. Er waren wel enkele onduidelijkheden in de beschrijving van de databank. Zo was onder andere de betekenis van `thal` niet helemaal duidelijk waardoor de validiteit hiervan niet helemaal verzekerd is. Er werd ook een veel hogere sensitiviteit verwacht voor het de door fysieke inspanning veroorzaakte angina, zo goed als 1. De lage sensitiviteit voor cholesterolgehalten groter dan 241 mg/dl klopt medisch gezien ook niet. Deze twee afwijkingen zijn rechtstreeks terug te brengen tot de slechte kwaliteit van de data. In de histogrammen valt op hoe gelijkvormig de verdelingen met - en zonder hartfalen zijn voor de cholesterolwaarden. Ook het niet optreden van een angina bij een inspanningsoefening treedt op bij bijna 40% van de patiënten met hartfalen. Een laatste tekortkoming in deze dataset is het feit dat er niet vermeld wordt of de patiënt al dan niet rookt, dit is een zeer belangrijke factor voor deze aandoeningen.

3.5 Besluit

De resultaten van het gewone naïef Bayesaans netwerk leveren, zover de data dit toelaat, medische correcte kennis. Door de strenge beperking die het gebruikte naïve Bayesaanse netwerk oplegt is het aannemelijk dat er in de data verbanden zijn met een hogere sensitiviteit die nu niet boven water gekomen zijn. In het volgende hoofdstuk zal dan ook worden nagegaan hoe een specifieker model gevonden kan worden, rekening houdend met de eerder gestelde beperkingen van berekenbaarheid.

Hoofdstuk 4

Generatie van specifieke kennis

4.1 Doelstelling

In dit tweede deel zal getracht worden specifiekere modellen op te stellen die een hogere sensitiviteit opleveren dan en gewoon naïef Bayesaans netwerk. De beperkingen van het vorige hoofdstuk blijven natuurlijk gelden. De kennis moet transparant zijn om gemakkelijk gevalideerd te kunnen worden door experten, stabiel zijn en bovenal geen complexe berekening vergen bij het decision support.

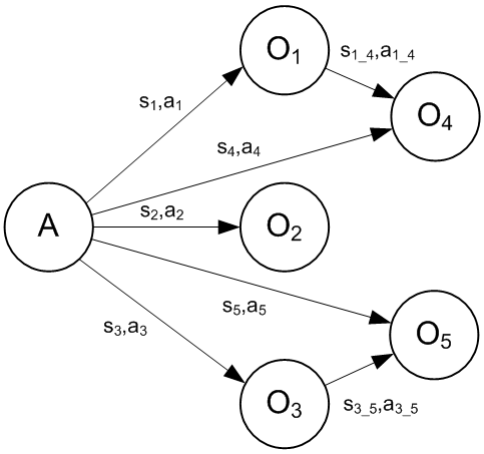
4.2 Modellering

Tree Augmented Naive Bayes

In de zoektocht naar een compromis tussen een beheersbare complexiteit en een exacter model werd ook *Tree Augmented Naive Bayes* (TAN)(Friedman et al. [9]) onderzocht. Een TAN is een superpositie van een naïef Bayesaans netwerk en een boomstructuur. Het resultaat is dat iedere observatie een kind is van de aandoening en daarenboven ook kind kan zijn van ten hoogste een andere observatie, zoals weergegeven in figuur 4.1. Het berekenen van de ideale boomstructuur, m.a.w. het selecteren van de belangrijkste edges kan gebeuren door verschillende scorefuncties zoals Minimum Description Length en Bayes score.

Hoewel de complexiteit van TAN veel kleiner is dan dat van gewone Bayesaanse netwerken, is de mogelijkheid van 2 parents voor een node nog altijd niet schaalbaar genoeg en wordt de voorkeur gegeven aan dynamische naïeve Bayesaanse netwerken.

Opmerkelijk was dat bij het experimenteren met TAN in Weka, bleek dat de gevonden modellen voor de dataset over hartfalen *niet* stabiel waren, dit in tegenstelling tot de resultaten beschreven in de literatuur [9]. Voor 5 tests, de totale dataset over hartfalen en iedere bron apart werden 5 totaal verschillende grafen gevonden. De gevonden structuren waren klinisch ook niet relevant. Op de cd-rom zijn enkele screenshots hiervan toegevoegd.



Figuur 4.1: Tree Augmented Naive Bayes

Dynamische Naïeve Bayesaanse Netwerken

Tot nu toe werden statische Bayesaanse netwerken gebruikt. Een dynamisch Bayesaans netwerk geeft aangepaste sensitiviteiten en aspecificiteiten naargelang de premisse. Zo kan een specifieker model gevonden worden in functie van een aanname bvb. een apart model per geslacht. Het netwerk kan ook variëren in de tijd. bvb. de hoeveelheid bloed vastgesteld op de CT-scan na een hersenbloeding neemt af naargelang de verstreken tijd. Op die manier kan men een model gebruiken dat aangepast is aan de fase waarin de aandoening is.

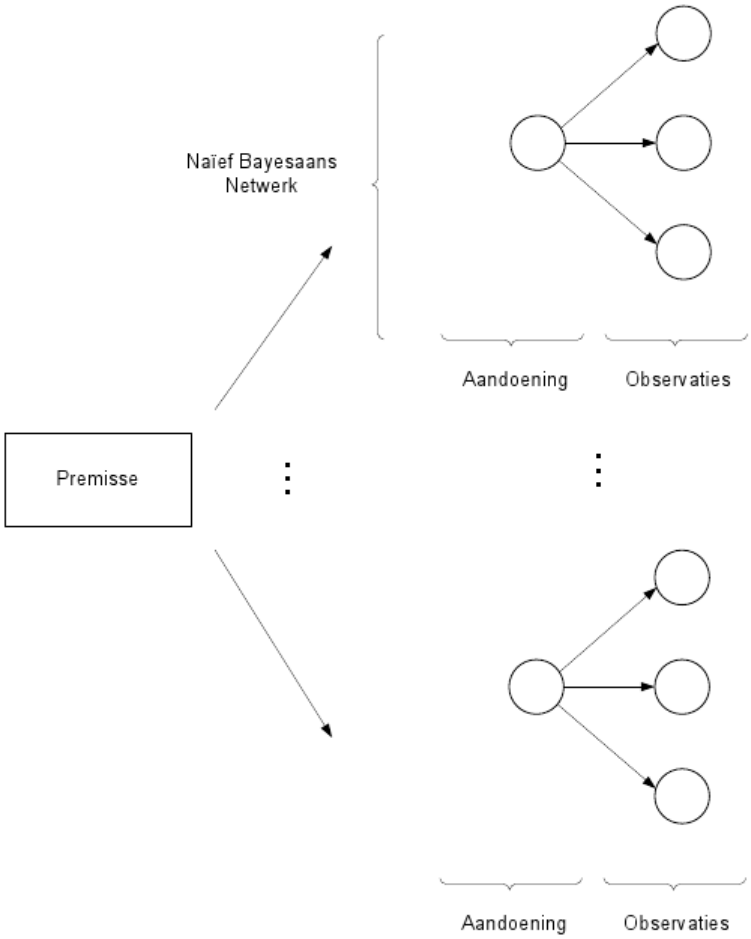
Door verschillende Bayesaanse netwerken te definiëren afhankelijk van een premisse, kunnen we tegelijkertijd een specifieker model toepassen terwijl de rekencomplexiteit bij het gebruik van deze kennis at runtime beheersbaar blijft. De prijs die we hiervoor betalen is dat de combinatorische explosie opschuift naar het aantal Bayesaanse netwerken dat zal opgeslagen worden. Deze explosie zal beperkt worden door naar de meest waardevolle premissen te zoeken (zie verder).

Een tweede manier waarop deze combinatorische explosie van het aantal modellen kan worden beperkt bij tijdsgebonden observaties is door eerst alle parameters per symptoom te berekenen in functie van de tijd (al dan niet gebruikmakend van klassen) waarna de resultaten hiervan benaderd worden met een functie. Verschillende regressie methoden kunnen hiervoor onderzocht worden. Logistische regressie (Hosmer and Stanely [10]) lijkt b.v.b. ideaal observaties te beschrijven die zich pa na verloop van tijd manifesteren.

De meest eenvoudige logistische functie is de sigmoide functie is weergegeven in figuur 4.3:

$$\sigma(t) = \frac{1}{1+e^{-t}} \tag{4.1}$$

Door een gebrek aan bruikbare data waaruit tijdsafhankelijkheid van symptomen afgeleid kan worden, is er enkel geëxperimenteerd met het verschuiven van observaties naar de premisse, m.a.w. het verschuiven van kinderen van de root naar de premisse. Voor



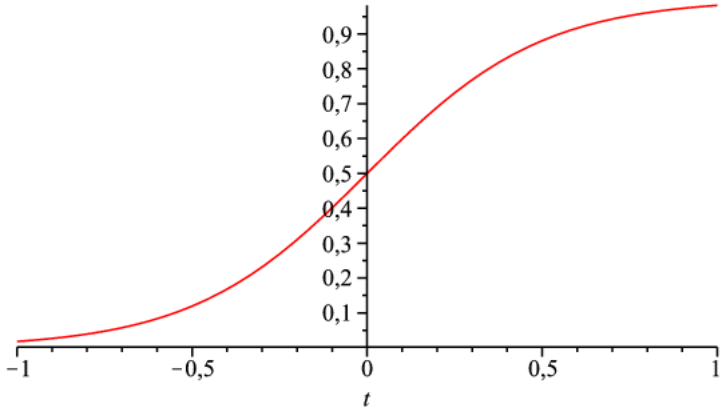
Figuur 4.2: Dynamisch naïef Bayesaans netwerk

het automatisch vinden van observaties die hiervoor in aanmerking komen, werd gebruik gemaakt van Toplog (zie verder).

4.3 Technologieën

4.3.1 Inleiding

Logisch programmeren werkt declaratief. In tegenstelling tot procedurele talen zoals Java, C, edm. moet men niet formuleren hoe men een resultaat wil bekomen, maar welk resultaat men wil bekomen. Hoewel in de abstract abductief logisch programmeren als een interessante mogelijkheid vermeld wordt, is er uiteindelijk toch geopteerd om een ILP-systeem (Toplog) om waardevolle premissen te vinden. Het vinden van theorieën door middel van een achtergrondkennis en voorbeelden, wat typisch is voor inductief



Figuur 4.3: Sigmoide Functie

logisch programmeren zal, zoals verder zal blijken, goed aansluiten bij de doelstellingen van deze thesis. Toplog is een supervised learning framework dat toelaat hypothesen te genereren wanneer een toplevelтheorie, achtergrondkennis en positieve en negatieve voorbeelden gegeven worden. De toplevelтheorie beschrijft welke vorm de gevonden hypothesen moeten hebben, zo wordt de zoekruimte van mogelijke hypothesen beperkt en is het vinden ervan haalbaar. De gevonden hypothesen zijn zinvolle premissen bij een relatie $observatie \rightarrow aandoening$. Hoe deze hypothesen gevonden worden en hoe deze uiteindelijk omgezet worden naar premissen wordt in de loop van dit hoofdstuk uitgelegd.

Een gedetailleerde beschrijving van het raamwerk is te vinden in het transfer report op de Toplog webpagina [23]. Als Prolog variant werd voor YAP gekozen vanwege de veel hogere performantie in vergelijking met het gekendere SWI-Prolog.

Eerst worden in dit hoofdstuk enkele begrippen uit logica en logisch programmeren uitgelegd. Er zal uitgegaan worden van clausal logic, de logica gebruikt door Prolog. Clausal logic en eerste orde predicaten logica zijn semantisch equivalent, iedere clausule in clausal logic kan vertaald worden naar een formule in eerste orde predicaten logica. Voor een veel diepgaandere bespreking wordt verwezen naar [8].

Syntax en semantiek

formeel kan men de syntax van *full clausal logic* als volgt beschrijven [19].

functor	:	single word starting with lower case
variable	:	single word start with upper case
term	:	variable | functor [(term[, term]*)]
predicate	:	single word starting with lower case
atom	:	predicate[(term[, term]*)]
clause	:	head[:-body]
head	:	[atom[; atom]*]
body	:	atom[, atom]*

Er zijn drie logische connectoren die atomen kunnen verbinden.

- conditie: `:-` (als)
- disjunctie: `;` (of)
- conjunctie: `,` (en)

Dit wordt verduidelijkt door het ontleden van enkele voorbeelden.

```
father(X,Y) :- parent(X,Y),male(X).
```

Wanneer X een ouder is van Y en X is mannelijk, dan is X vader van Y, dit geldt voor alle waarden van de variabelen `X` en `Y`. Merk op dat de antecedent rechts staat van het `if`-statement, en het gevolg links. `father(X,Y)`, `parent(X,Y)` en `male(X)` zijn drie atomen met elk een ariteit van respectievelijk 2, 2 en 1.

Een clausule zonder body is onvoorwaardelijk waar en wordt een feit genoemd. bvb.:

```
male(polonius).
parent(polonius, ophelia).
```

In full clausal logic is het toegelaten om een disjunctie in de head van een clausule te hebben. In Prolog, waar men gebruik maakt van Horn-clausules is dit echter niet toegelaten. Horn clausules zijn clausules met maximum één atom in de head. De volgende regel is dus legaal in full clausal logic, maar niet in definite clausal logic.

```
father(X,Y); mother(X,Y) :- parent(X,Y).
```

Proof Theory

Met de voorgaande regel en feiten kan Prolog de volgende vragen beantwoorden:

```
?- father(X,ophelia).
  X = polonius.
```

```
?- father(polonius,X).
  X = ophelia.

?- father(polonius,ophelia).
  true.
```

Om oplossingen (resolutions) te vinden gebruikt Prolog het principe van een *bewijs uit het ongerijmde* (reductio ad absurdum). Iets wordt bewezen in Prolog door SLD-resolution. SLD staat voor *Selective Linear Definite clause resolution* en werd geïntroduceerd door Kowaslki [14],[13]. Het vinden van antwoorden komt overeen met het doorlopen van de SLD-boom. De root van de boom is de query. De kinderen van de nodes zijn het resultaat van substituties in de ouder.

Voor ons eenvoudig voorbeeld bestaat de SLD-boom slechts uit één succesvolle tak, dit is iets te triviaal om het concept van SLD-resolotie op een voldoende wijze te demonstreren. Daarom is er in de bijlage is een iets complexer voorbeeld toegevoegd uit [8].

4.3.2 Werking van Toplog

Inleiding

Het doel van Inductief logisch programmeren is om met een gegeven eindige verzameling van predicaten B, de achtergrondkennis en twee bijkomende verzamelingen van clausules E^+ met positieve voorbeelden en E^- met negatieve voorbeelden een theorie Σ te vinden waarvoor geldt dat

$$\forall e^+ \in E^+ : \Sigma \cup B \models e^+ \tag{4.2}$$

$$\forall e^- \in E^- : \Sigma \cup B \not\models e^- \tag{4.3}$$

$\Sigma \cup B$ rekening houdend met E^+ en E^-. Een triviale maar nietszeggende theorie $\Sigma = E^+$ is correct, maar waardeloos. Ieder nieuw voorbeeld dat niet voorkwam in E^+ zal als negatief geclassificeerd worden. Het vinden van een theorie met een *minimum description length (*MDL*)* die zo dicht mogelijk een correcte theorie benadert, leidt tot meer bruikbare theorieën. ILP-systemen houden dan ook niet enkel rekening met hoe goed een theorie de positieve en negatieve voorbeelden classificeert, maar ook met de lengte van de theorie.

TDHD-framework

TDHD is een framework day het zoeken naar mogelijke hypothesen versnelt door de te doorzoeken hypostheseruimte te verkleinen. Het beperken van de zoekruimte gebeurt door de vorm van de te vinden hypotheses mee te geven als input aan het ILP-systeem. Toplog gebruikt als inputvector $S_{TDHD} = \langle NT, \top, B, E \rangle$ waar NT een verzameling "non terminal"predicaat symbolen voorafgegaan door een `$`-teken. $\top$ is een logisch programma dat de declaratieve bias over de hypotheseruimte vertegenwoordigt, B en E zijn de reeds eerder besproken achtergrondkennis en voorbeelden (zowel positieve als negatieve). Iedere

clausule in $\top$ bevat minstens één element uit NT terwijl er in B en E geen enkel element uit NT mag voorkomen.

De bedoeling van TDHD is het vinden van een consistente verzameling van hypothese clausules H, zonder elementen uit NT waarvoor iedere element $h \in H$ er op zijn minst één positief voorbeeld $e \in E$ gevonden kan worden, waarvoor geldt:

$$\top \models h \tag{4.4}$$

$$B, h \models e \tag{4.5}$$

In deze thesis is gebruik gemaakt van mode declarations om de hypostheseruimte te beschrijven. Mode declarations worden door Toplog automatisch omgezet naar een $\top$-theorie. `modeh` en `modeb` beschrijven de vorm die een head en respectievelijk de body van een hypothese $h \in H$ kan aannemen.

Bijvoorbeeld:

```
modeh(observatie_i(+patient)).
modeb(observatie_j(+patient)).
modeb(aandoening_a(+patient)).
```

wordt vertaald naar een $\top$-theorie:

```
⊤1 : observatie_i(X) ← $body(X).
⊤2 : $body(X) ← . %een empty body
⊤3 : $body(X) ← observatie_j(X).
⊤4 : $body(X) ← aandoening_a(X).
```

De verzameling NT wordt hier beschreven door $NT = \$body$

De achtergrondkennis B is een opsomming van feiten over patiënten:

```
observatie_j(patient_x).
observatie_j(patient_y).
aandoening_a(patient_y).
...
```

Tot slot worden ook nog de positieve en negatieve voorbeelden meegegeven, het al dan niet vaststellen van de observatie uit de head per patient. Bij positieve voorbeelden wordt een positief getal meegegeven als parameter, bij negatieve een negatief. De absolute waarde van dat getal is het gewicht dat aan dat voorbeeld gegeven wordt. Hier wegen alle patiënten even zwaar door.

```
example(observatie_i(patient_x),1).
example(observatie_i(patient_y),-1).
...
```

Top Directed Hypothesis Derivation

Het afleiden van hypothesen gebeurt in twee stappen. In een eerste stap wordt een voorbeeld e bewezen door middel van de achtergrondkennis B en de $\top$-theorie. De $\top$-theorie wordt uitgevoerd met het voorbeeld e als beginclausule. Deze uitvoering houdt het bewijs in bestaande uit een reeks clausules van de $\top$-theorie en de achtergrondkennis B.

Zo kan door gebruik te maken van de reeds eerder beschreven $\top$-theorie en de volgende clausule uit de achtergrondkennis $b_1 =$ `observatie_j(patient_x)`. Het verwerpen van de negatie van het willekeurige voorbeeld $e_1 =$ `example(observatie_i(patient_x),1)` leidt tot deze weerleggingen: $r_1 = \langle \neg e_1, \top_1, \top_2 \rangle$ en $r_2 = \langle \neg e_1, \top_1, \top_3, b_1, \top_2 \rangle$.

In een tweede stap worden door weerleggingen R herschikt naar de vorm $R' = D_h R_e$ waar D_h een SLD-oplossing is van de hypothese h waar (4.4) en (4.5) voor gelden. Een formeel bewijs hiervoor wordt gegeven in [23]. Door SLD-resolutie toe te passen op r_1 en r_2 worden de volgende clausules afgeleid:

- $h_1 =$`observatie_i(X)` met $d_{h1} = \langle \top_1, \top_2 \rangle$
- $h_1 =$`observatie_i(X)` $\leftarrow$ `observatie_j(patient_x)` met $d_{h2} = \langle \top_1, \top_3, \top_2 \rangle$

4.4 Implementatie

Aanpassing Java-programma

Om nodige input-bestanden voor Toplog te genereren werd een bijkomende module `ToplogConverter` geschreven. Deze module schrijft per observatie een $S_{TDHD} = \langle NT, \top, B, E \rangle$-vector zoals hierboven beschreven. De $\langle NT, \top \rangle$ wordt, zoals eerder beschreven impliciet meegegeven door modedeclarations te gebruiken.

`ToplogConverter` gebruikt dezelfde parameters als de module om een naïef Bayesaans netwerk te genereren maar vereist daarenboven ook:

- Een verwijzing naar de kolom met de id's.
- De stamnaam voor de Toplog-bestanden. Indien de stamnaam `temp` is zullen bestanden aangemaakt worden als `temp_age.pl`, `temp_chol.pl` enz, voor iedere relevante kolom.

4.4.1 Toplog als hulpmiddel om waardevolle premissen te vinden

Het naïeve Bayesaanse netwerk gebruikte tot nu toe slechts regels die er als volgt uitzien $P(observatie_i | aandoening_a)$, gekoppeld met een sensitiviteit s en een aspecificiteit a. Dezelfde regel ziet er in Prolog als volgt uit:

```
observatie_i(X) :- aandoening_a(X).
```

Met behulp van de inputfile besproken in de vorige sectie, genereert Toplog theorieën met regels die er als volgt uitzien:

```
observatie_i(X) :- aandoening_a(X), observatie_j(X).
```

M.a.w.: `observatie_i` zal waargenomen worden indien `aandoening_x` *en* `observatie_j` waargenomen zijn. Bijvoorbeeld:

```
trestbps(X) :- heartdisease(X), age(X).
```

Patiënt X behoort tot de risicogroep qua aantal hartslagen per minuut in rust ($>= 132bps$) *als* één of meerdere bloedvaten meer dan 50% verstopt is *én* de patiënt qua leeftijd ook tot de risicogroep behoort (ouder dan of gelijk aan 55 jaar). Met sensitiviteit 0,74 en aspecificiteit 0,38.

De regel uit Toplog kunnen we herschrijven als:

$$P(observatie_i|aandoening_a, observatie_j)$$

of:

$$observatie_j \Rightarrow P(observatie_i|aandoening_a)$$

of nog, in N3:

```
{:observatie_j e:boolean e:T} =>
  {({:observatie_i e:boolean e:T}{:aandoening_a e:boolean e:T}) e:conditional s}
{:observatie_j e:boolean e:T} =>
  {({:observatie_i e:boolean e:F}{:aandoening_a e:boolean e:F}) e:conditional a}
```

Wanneer `observatie_j` *waargenomen is, dan geldt* $P(observatie_i|aandoening_a)$ *met sensitiviteit s en aspecificiteit a.* Doordat de door Toplog gevonden regels met meer factoren rekening houden zijn die specifieker, en hebben die in het algemeen een hogere sensitiviteit dan de veel algemenere regels die slechts met één observatie rekening houden.

4.5 Evaluatie

Door gebruik te maken van de door Toplog gevonden premissen worden zoals verwacht hogere sensitiviteiten gevonden. Het valt ook op dat de aspecificiteiten gestegen zijn. Dit valt te verklaren door de achtergrondkennis B die enkel positieve statements bevat. Dit resulteert in iets meer vals positieven dan vals negatieven, of nog, een lichte bias naar de rechterbovenhoek van de ROC-curve. Desondanks zijn deze resultaten aanvaardbaar voor Agfa.

Voor bijna alle observaties is er minstens een premisse gevonden waarvoor significant hogere sensitiviteit gevonden werd. De enige uitzondering hierop was de door inspanning geïnduceerde angina. Dit was de observatie met de hoogste sensitiviteit in het vorige hoofdstuk, toen er geen premissen gebruikt werden. Een neveneffect was de stijging van

de aspecificiteiten, die ook de MCC doen dalen. Toch zijn bij slechts vier observaties enkel premissen gevonden waarbij de stijging van de sensitiviteit een zodanige stijging van de aspecificiteit met zich meebracht zodat de MCC ook (licht) daalde. Bij een van deze observaties, `slope=2`, werd wel een andere regel gevonden waarbij de aspecificiteit meer gereduceerd werd dan de sensitiviteit, waardoor de MCC gestegen was tegenover het resultaat zonder premisse.

CONDITIE	PREMISSE	S	A	MCC
age >= 55	restecg>=1	0.74	0.39	0.36
age >= 55	sex=1	0.76	0.51	0.26
age >= 55	thal=3	0.45	0.18	0.3
trestbps >= 132	age>=55	0.73	0.66	0.09
chol >= 241	slope =2 AND exang = 1	0.91	0.77	0.19
chol >= 241	slope =2	0.79	0.64	0.17
chol >= 241	exang = 1	0.85	0.76	0.12
thalach <142	sex=1	0.78	0.43	0.35
thalach <142	exang=1	0.88	0.71	0.21
oldpeak >= 0.89	exang=1	0.89	0.72	0.2
ca >= 1	age>=55	0.81	0.33	0.49
ca >= 1	chol>=241 AND exang=1	0.92	0.72	0.27
ca >= 1	sex=1	0.85	0.33	0.52
sex=1	cp=4	0.83	0.56	0.24
sex=1	age>=55	0.76	0.43	0.28
sex=1	ca>=1	0.85	0.47	0.38
exang=1	thalach <142 AND slope =2	0.9	0.77	0.19
exang=1	thalach <142 AND oldpeak >=0.89	0.9	0.71	0.24
exang=1	cp=4 AND oldpeak >=0.89	0.93	0.77	0.23
cp=4	sex=1	0.83	0.36	0.49
cp=4	oldpeak>=0.89	0.89	0.45	0.47
restecg=1	age >=55	0.8	0.67	0.13
slope=2	oldpeak>=0.89	0.82	0.66	0.18
slope=2	sex=1 AND age>=55	0.85	0.75	0.13
slope=2	cp=3	0.51	0.2	0.34
thal=7	sex=1	0.81	0.49	0.33
thal=7	oldpeak>=0.89	0.91	0.47	0.49

Tabel 4.1: Resultaten van naïeve Bayes met premissen

Validatie

Bij de validatie van de resultaten door een arts van Agfa HealthCare werden geen discrepanties tussen de data en de resultaten gevonden. De grenswaarde voor een te

hoge bloeddruk stijgt naargelang de leeftijd toeneemt, hoe ouder iemand is, hoe hoger de toegelaten bloeddruk. Dit is een verband dat door de gebruikte methode voorlopig onmogelijk te vinden is omdat de patiënten eerst worden opgedeeld in risicogroepen, waarna er naar onderlinge afhankelijkheden gezocht wordt.

4.6 Besluit

Door gebruik te maken van premissen worden specifiekere Bayesaanse netwerken gevonden. Als rechtstreeks gevolg hiervan zijn de sensitiviteiten significant hoger dan bij de gewone Bayesaanse netwerken. In de meeste gevallen werd ook een stijging van de MCC vastgesteld, wat wijst op een algemene stijging van de kwaliteit als classifier voor die bepaalde subset.

Bij de validatie kwam aan het licht dat de gebruikte methode de afhankelijkheid van de thresholdwaarden van een observatie van een andere observatie niet kan vinden. Hoewel dit met deze beperkte dataset sowieso onmogelijk is om zo'n verband te vinden, zal indien dit programma in de praktijk gebruikt wordt hiermee rekening moeten houden.

Hoofdstuk 5

Algemeen besluit

5.1 Verwezenlijkingen

In dit eindwerk werden twee verschillende soorten bijkomende kennisbronnen gegenereerd. Een algemeen naïef Bayesaans netwerk dat geldt voor alle observaties uit de dataset en specifiekere Bayesaanse netwerken afhankelijk van premissen. Dit algemene model is toepasbaar voor alle observaties, de gevonden kennis kan dus in principe gebruikt worden om de kans op hartfalen te bepalen. Indien een van de premissen uit de resultaten van het tweede deel vervuld is, kunnen specifiekere waarden voor s en a gebruikt worden. Het resultaat hiervan zal een diagnose zijn met een hogere kwaliteit dan wanneer er geen premissen gebruikt worden.

Omdat bij medische toepassingen foute diagnoses absoluut vermeden moeten worden is het noodzakelijk dat resultaten van datamining altijd gevalideerd worden door een arts. Datamining kan een handige tool zijn om kennisbanken aan te maken, dit kan om de zojuist vernoemde reden echter nooit volledig geautomatiseerd gebeuren.

Men kan besluiten dat de doelstellingen van het eindwerk behaald zijn, maar er blijft nog veel ruimte over voor toekomstig onderzoek en eventuele verbeteringen.

5.2 Verder onderzoek

5.2.1 Mogelijk verbeteringen en uitbreidingen

Heuristieken

Er kunnen alternatieve heuristieken gezocht worden voor het binariseren. Hoewel het niet voorkwam in de gebruikte dataset is het mogelijk dat de verdeling van patiënten met een aandoening niet benaderd kan worden met een normale verdeling. Wanneer een aandoening bijvoorbeeld veel voorkomt bij jonge kinderen en ouderen, volstaat een gewone binarisatie niet. Hiervoor is een heuristiek noodzakelijk die in zulke gevallen het bereik van de observatie opsplitst in meerdere klassen en de klassen met een hoger risico voor

een aandoening identificeert. Het is nu wel al mogelijk om in het programma manueel een klasse te definiëren en daarvoor de bijhorende parameters te laten berekenen.

Een tweede uitdaging is om een heuristiek te vinden die afhankelijkheid van de thresholdwaarde nagaat in functie van andere observaties, zoals de leeftijdsgebonden grenswaarde voor de bloeddruk. Het vinden van zulke vrij subtiele verbanden zal een grote hoeveelheid kwalitatieve data vereisen. Zo moet er voor een representatief aantal leeftijdscategorieën een bloeddruk threshold gevonden worden met een vrij kleine errormarge vooralleer men kan besluiten dat er daadwerkelijk zo'n verband is. Omdat een exhaustieve berekening van alle mogelijke verbanden in de praktijk waarschijnlijk ondoenbaar is, zal misschien toevlucht genomen moeten worden tot metaheuristieken, of de zoekruimte beperken door expertenkennis te gebruiken, b.v.b. een arts kan expliciet aanduiden welke observaties potentieel interessante correlaties bevatten (zoals cholesterol thresholdwaarden en leeftijd). Deze beperktere zoekruimtes zijn waarschijnlijk wel volledig berekenbaar.

Automatisch aansturen van Toplog

Tot nu toe worden enkel de inputfiles voor Toplog automatisch gegenereerd. De regels uit de door Toplog gevonden theorieën moeten nog manueel aan het Java-programma gegeven worden om de sensitiviteiten en aspecificiteiten te berekenen voor de volledige dataset en eventueel N3-regels te genereren. De sensitiviteit en de aspecificiteit van de door Toplog gevonden theorieën worden nu wel reeds afgedrukt op de `stdout` voor zowel de test- als de trainingset.

Vertaling van kolomnaam naar namespace

Bij het schrijven van N3-regels werden nu de kolomnamen gebruikt. Het vertalen van de kolomnamen naar in het semantische web relevante *qname*'s viel buiten de scope van deze thesis, maar zal evenwel belangrijk zijn om de gegenereerde kennis in de praktijk te kunnen gebruiken.

Folding

Hoewel de gevonden resultaten sowieso gevalideerd moeten worden door een expert binnen het medische domein, kan het opdelen van de data in een trainingset en een testset helpen om de kwaliteit van de gevonden resultaten na te gaan. Wanneer de kennis, gevonden uit de trainingset gelijkaardige resultaten geeft op een testset, is het waarschijnlijk dat de geleerde kennis ook toepasbaar zal zijn op andere nieuwe data. Dit principe wordt ook gebruikt bij k-folding [26], waar een dataset verdeeld wordt in K delen. K-1 delen worden gebruikt als trainingset, en 1 deel als testset. Dit wordt herhaald voor elk van de K subsets waarna de resultaten worden uitgemiddeld. Zo worden schommelingen afhankelijk van de gunstige of minder gunstige opdeling in test en trainingset uitgemiddeld.

5.2.2 Alternatief

Het kan interessant zijn om in de toekomst *fuzzy cognitive maps* (FCM) te onderzoeken als alternatief voor Bayesaanse netwerken. Fuzzy cognitive maps werden voor het eerst geïntroduceerd door Kosko [12]. Het is een soft-computing techniek voor het modelleren van complexe systemen door een directionele graaf die oorzaken en gevolgen verbinden. Een voordeel van FCM's is het gemak waarmee expertenkennis eraan toegevoegd kan worden.

De bruikbaarheid voor medical decision support systems werd beschreven in [24] en [20].

Bibliografie

[1] Wikipedia, the free encyclopedia. `http://www.wikipedia.org`.

[2] David W. Aha. UCI machine learning repository, heart disease databases, 1988.

[3] Tim Berners-Lee. Primer: Getting into rdf & semantic web using n3.

[4] Tim Berners-Lee and Dan Connolly. Notation3 (N3): A readable rdf syntax. W3C team submission, W3C, January 2008.

[5] Remco R. Bouckaert. Bayesian network classifiers in weka. Technical report, 2004.

[6] Christopher P. Cannon. *Management of Acute Coronary Syndromes, 2nd Edition*. Humana Press, 2002.

[7] Helen Chen and Jos De Roo. Use case: Using semantic web and proof technologies to reduce errors in radiological procedure orders. February 2007.

[8] Peter Flach. *Simply Logical*. John Wiley, April 1994.

[9] Nir Friedman, Dan Geiger, and Moises Goldszmidt. Bayesian network classifiers. *Mach. Learn.*, 29(2-3):131–163, 1997.

[10] David W. Hosmer and Stanley Lemeshow. *Applied logistic regression (Wiley Series in probability and statistics)*. Wiley-Interscience Publication, September 2000.

[11] De Winne Keppens, Ryckaert. *Statistiek*. KaHo Sint-Lieven, 2008.

[12] B. Kosko. Fuzzy cognitive maps. *International Journal of Man-Machine Studies*, (24):65–75, 1986.

[13] Robert Kowalski. Predicate logic as programming language. *Proceedings IFIP Congress 1974*, pages 569–574, 1974.

[14] Robert Kowalski and Donald Kuehner. Linear resolution with selection function. *Artificial Intelligence*, 2:227–260, 1971.

[15] Daniel T. Larose. *Discovering knowledge in data : an introduction to data mining*. Wiley-Interscience, Hoboken, N.J., 2005.

[16] Daniel T. Larose. *Data Mining Methods and Models*. Wiley-IEEE Press, January 2006.

[17] G. Logghe. *Farmacologie en -Dynamie Deel 2*. KaHo Sint-Lieven, 2003.

[18] Christian Lovis, Dirk Colaert, and Veli N Stroetmann. Debugit for patient safety - improving the treatment with antibiotics through multimedia data mining of heterogeneous clinical data. *Stud Health Technol Inform*, 136:641–646, 2008.

[19] Joris Maervoet. Logic programming, an introduction to prolog, logic programming and applications, 2009.

[20] E. I. Papageorgiou, P. P. Spyridonos, D. Th. Glotsos, C. D. Stylios, P. Ravazoula, G. N. Nikiforidis, and P. P. Groumpos. Brain tumor characterization using the soft computing technique of fuzzy cognitive maps. *Appl. Soft Comput.*, 8(1):820–828, 2008.

[21] M. Pyckavet. *Fysiologie Pathologie*. KaHo Sint-Lieven, 2003.

[22] Irina Rish. An empirical study of the naive bayes classifier. In *IJCAI-01 workshop on Empirical Methods in AI*.

[23] José Carlos Almeida Santos. Toplog: Ilp using a logic program declarative bias. TRANSFER REPORT. 2008.

[24] Chrysostomos D. Stylios, Voula C. Georgopoulos, Georgia A. Malandraki, and Spyridoula Chouliara. Fuzzy cognitive map architectures for medical decision support systems. *Applied Soft Computing*, 8(3):1243 – 1251, 2008. Forging the Frontiers - - Soft Computing.

[25] Weka Machine Learning Project. Weka. URL http://www.cs.waikato.ac.nz/˜ml/weka.

[26] Ian H. Witten and Eibe Frank. *Data Mining: Practical Machine Learning Tools and Techniques, Second Edition (Morgan Kaufmann Series in Data Management Systems)*. Morgan Kaufmann, June 2005.

Bijlage A

Notation3

*Notation3 (*N3*)* is een taal ontworpen voor het semantische web. Het implementeert het *Resource Description Framework (*RDF*)* volledig en heeft daarenboven nog extra mogelijkheden zoals het uitdrukken van logische regels en het verwijzen naar N3 regels binnen N3 regels. De relatie tussen N3, zijn subsets en RDF zijn weergegeven in figuur A.1. Het *Semantische Web* is gebaseerd op informatie die beschreven is in RDF. RDF beschrijvingen in N3 zijn over het algemeen veel korter en leesbaarder dan in XML. Voor een veel diepgaandere studie wordt verwezen naar [3] en [4].

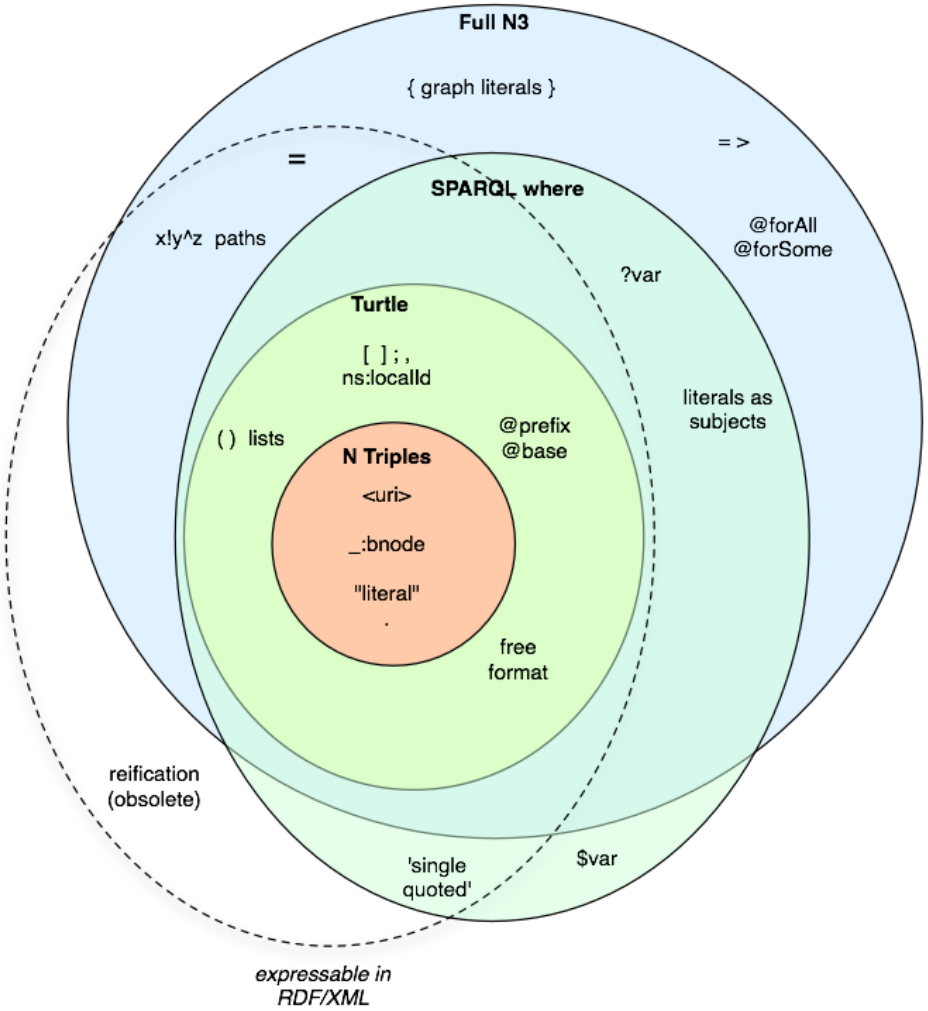
RDF informatie is een opsomming van feiten met een *onderwerp, werkwoord en object*. De tripple `x p y` beschrijft de relatie `p` tussen `x` en `y`. Net als bij RDF/XML zijn er ook namespaces waardoor ieder predicaat uniek geindetificeerd kan worden. De karakters voor het dubbelpunt, de prefix zijn afkortingen voor de namespace `uri`'s, zo is de volledige *qname* (bvb. `x:firstname`) uniek gedefinieerd. Aan deze predicaten kan ook een eigenschap worden toegeschreven zoals `[ x:firstname "Ora"]` betekent *iemand met de voornaam "Ora "*.

```
@prefix x: <http://example.org/x-ns/>.
@prefix dc: <http://purl.org/dc/elements/1.1/>.
@prefix n3: <http://www.w3.org/2004/06/rei#>.

 { [ x:firstname "Ora" ] dc:wrote [ dc:title "Moby Dick" ] } a n3:falsehood .
```

Tussen de accolades staat de tripple *Er is iemand met de voornaam "Ora "die een boek geschreven heeft met de titel "Moby Dick "*. Hier is `[ x:firstname "Ora"]` het onderwerp, `dc:wrote` het werkwoord en `[ dc:title "Moby Dick"]` het object.

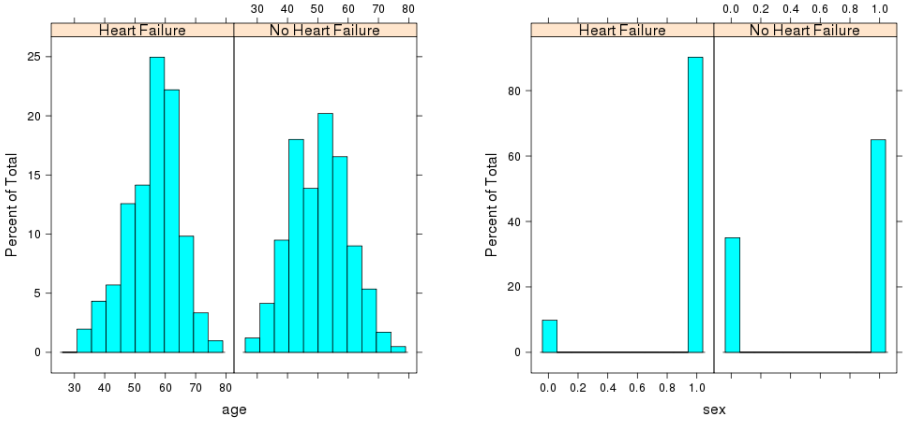
`a n3:falsehood.` geeft aan de tripple tussen de accolades onwaar is. Waardoor de volledige regel in het Nederlands betekent: *Er is niemand met de voornaam "Ora "die een boek geschreven heeft met de titel "Moby Dick "*.

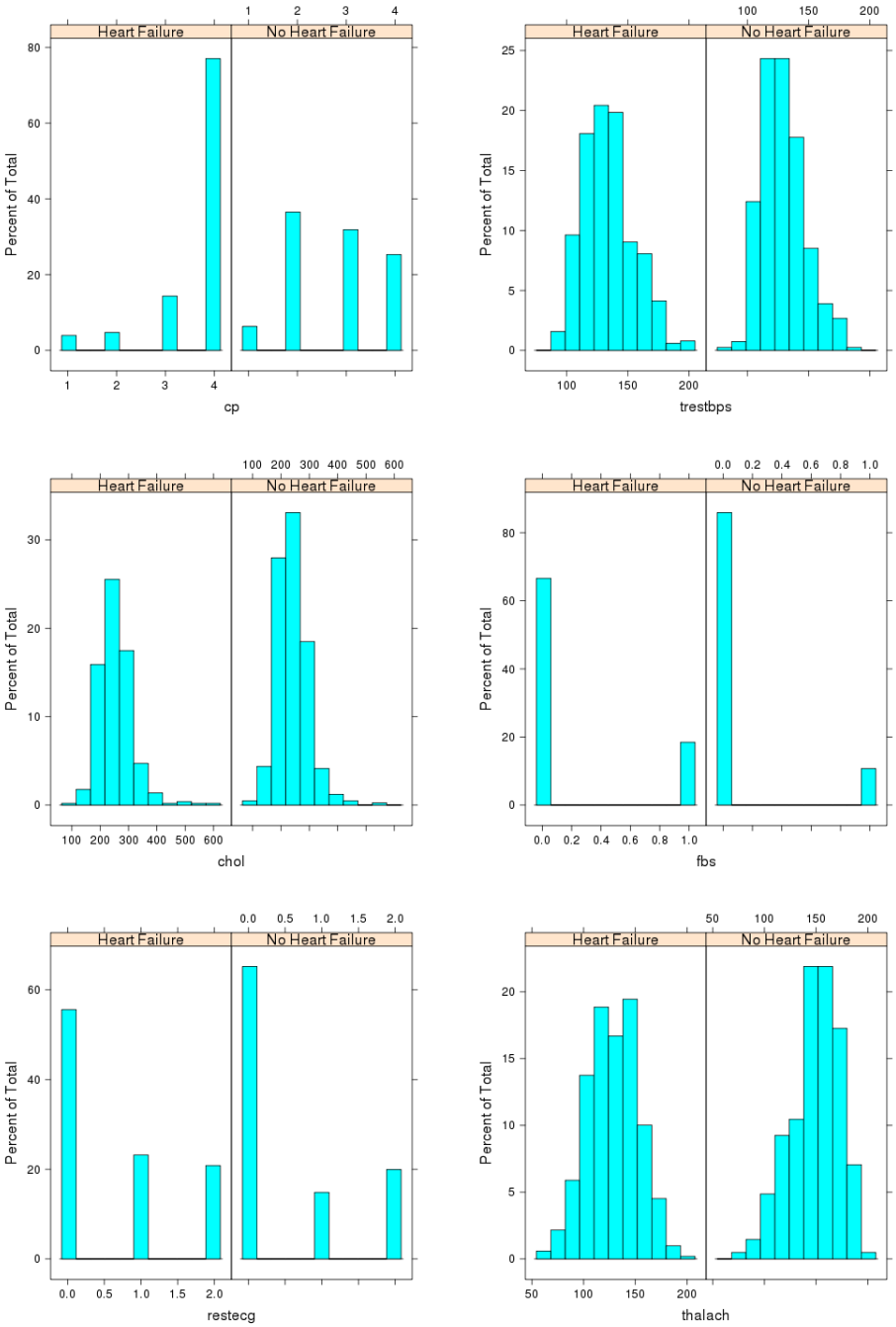


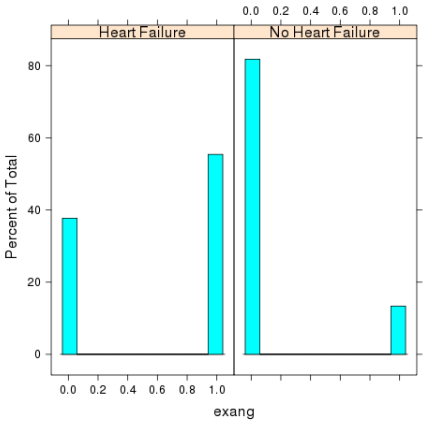
Figuur A.1: N3 subsets

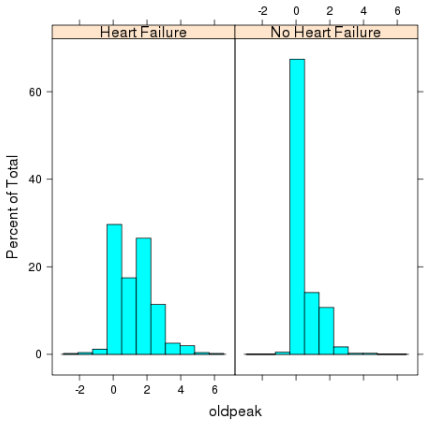
Bijlage B

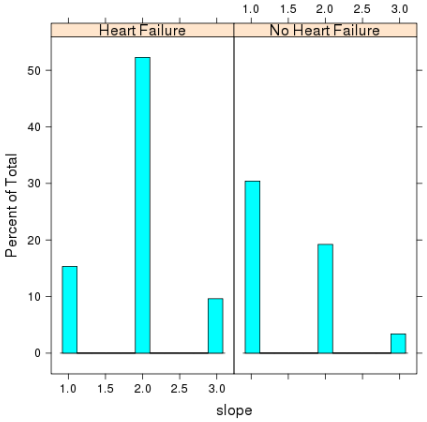
Historgrammen UCI Heart Disease Dataset

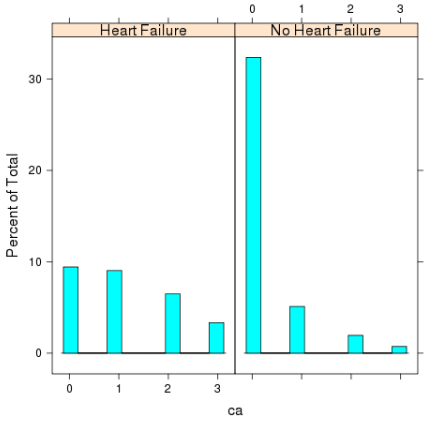


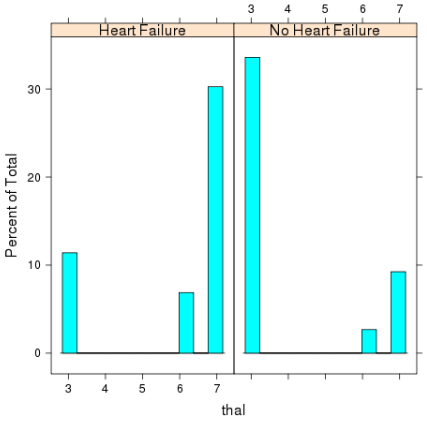

Heart Failure
No Heart Failure
Percent of Total
cp
trestbps
chol
fbs
restecg
thalach


Heart Failure
No Heart Failure
Percent of Total
exang


Heart Failure
No Heart Failure
Percent of Total
oldpeak

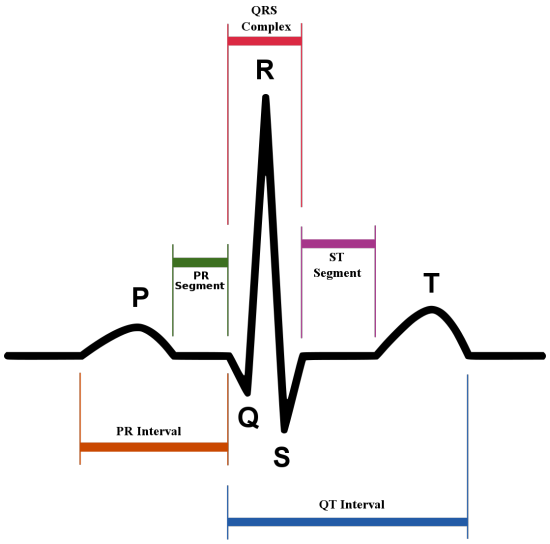

Heart Failure
No Heart Failure
Percent of Total
slope


Heart Failure
No Heart Failure
Percent of Total
ca


Heart Failure
No Heart Failure
Percent of Total
thal

Bijlage C

Elektrocardiogram



Figuur C.1: Elektrocardiogram

Een elektrocardiogram is een plot van de elektrische activiteit van het hart. Een diepgaande bespreking hiervan valt buiten de scope van deze thesis.

De *slope* in de dataset geeft de potentiaalverschuiving weer van het ST-segment. Een vlak ST-segment is normaal. Een ST-depressie wijst op een onevenwicht tussen de zuurstofvraag van het hart en de zuurstoftoevoer. De kransslagaders staan niet voldoende open waardoor de hartspier niet voldoende zuurstof krijgt. Dit wordt meestal veroorzaakt door atherosclerose (toeslibben van de aders). Een ST-elevatie wordt veroorzaakt door een acuut myocard infarct, m.a.w. een hartinfarct. Dit is een acute necrose van de hartspier door onvoldoende bloedtoevoer. De oorzaak is meestal de acute afsluiting van een coronaire arterie (kransslagader) door trombose (bloedstolsel) als gevolg van atherosclerose. [17] en [21]

Bijlage D

SLD-resolution

Het begrip SLD-resolution wordt uitgelegd met een voorbeeld uit [8]. Gegeven het volgende logische programma.

```
student_of(X,T):-follows(X,C),teaches(T,C).
follows(paul,computer_science).
follows(paul,expert_systems).
follows(maria,ai_techniques).
teaches(adrian,expert_systems).
teaches(peter,ai_techniques).
teaches(peter,computer_science).
```

Om te weten te komen wie de studenten van Peter zijn wordt deze query ingevoerd:

```
?-student_of(S,peter)
```

Deze query is equivalent met `:-student_of(S,peter)`, wat betekent *Peter heeft geen studenten*. Wanneer Peter wel studenten heeft, zal dit blijken wanneer we de query proberen te bewijzen. Er wordt een contradictie gevonden wanneer het bewijs leidt tot een lege regel. Deze lege regel wordt voorgesteld door $\square$. In dit voorbeeld zal de query twee maal verworpen worden door de antwoorden: $\{$`S` $\rightarrow$ `paul`$\}$ en $\{$`S` $\rightarrow$ `maria`$\}$.

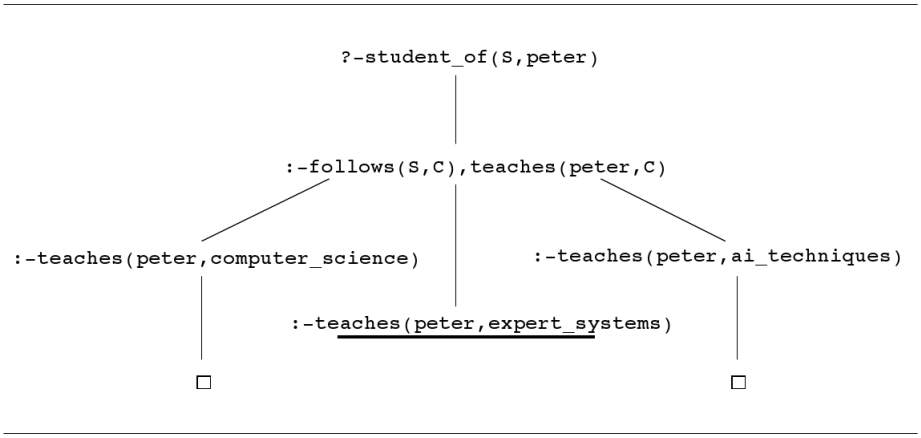
Aan de hand van de eerste clausule kunnen we besluiten dat de query gelijk is aan:

```
:-follows(S,C),teaches(peter,C)
```

Nu zoekt Prolog eerst naar een clausule met `follows(S,C)` als head, merk op dat dit ook een *feit* kan zijn, dit is immers een clausule die onvoorwaardelijk waar is, of nog een clausule waar de body `true` is. De eerste clausule die daaraan voldoet is `follows(paul,computer_science)`, daaruit volgt $\{$`S` $\rightarrow$ `paul, C` $\rightarrow$ `computer_science`$\}$. Nu wordt er gezocht naar een clausule `teaches(peter,computer_science)`. Deze clausule wordt gevonden op de laatste regel van het logische programma. Dit leidt tot een de lege regel $\square$, waarmee `:-teaches(peter,computer_science)` verworpen wordt en bewezen wordt dat Paul een student is van Peter. Nu wordt analoog de derde clausule geprobeerd, dit leidt eerst tot

$\{$`S` $\rightarrow$ `paul, C` $\rightarrow$ `expert_systems`$\}$. De clausule `teaches(peter,expert_systems)` komt echter niet voor in het logische programma, waardoor deze tak dood loopt. Het tweede antwoord wordt volledig analoog als het eerste gevonden en is weergegeven in D.1.

Prolog probeert de literals van links naar rechts te unificeren, dus indien de eerste clausule `student_of(X,T):-teaches(T,C),follows(X,C).` was geweest, zou er eerst naar een clausule gezocht worden waarmee `teaches(T,C)` geünificeerd kan worden.



Figuur D.1: Een voorbeeld van SLD-resolutie

Bijlage E

Inhoud van de bijgevoegde cd-rom

De bijgevoegde cd-rom bevat:

- De programmacode.
- De MySQL-dump en csv-bestanden van de gebruikte dataset.
- Een volledig klassendiagramma.
- De output van de resulaten.
- Enkele screenshots van de TAN netwerken.
- Dit einwerk in pdf-formaat.
- De poster en het wetenschappelijke artikkel.

Bijlage F

Beschrijving van deze masterproef in de vorm van een wetenschappelijk artikel

Katholieke Hogeschool Sint-Lieven
Industrial Engineering Departement
Electronics-ICT
Option Information and Communication Technologies
Gebroeders Desmetstraat 1, 9000 Gent

Generation of medical knowledge from real-life data

Author: Boris De Vloed
Supervisor: dr. ir. Annemie Vorstermans
Cosupervisor: lic. ing. Joris Maervoet
External supervisor: ir. Jos De Roo

Academic year 2008 - 2009

Abstract

Naive Bayesian networks are commonly used for their great performance/-complexity trade-off. In this article the adaptable naive Bayesian networks are compared to the more general common naive Bayesian networks. Adaptable naive Bayesian networks combine the low computational impact from common naive Bayesian networks, but allow a higher performance for classification with more specialised networks. The ilp-framework Toplog will be used for the construction of the adaptable networks. The performance of both approaches will be compared. The adaptable Bayesian networks show a significantly higher sensitivity when certain premises are met. Due to a bias in the way the adaptable networks are formed, the aspecificity slightly increases too.

1 Introduction

Medical decision support systems MDSS advise clinical professionals on how sure they can be about a certain diagnosis. Additionally it indicates how much confidence can be gained about a diagnosis when certain additional tests are done and what the costs of those extra tests are. Costs include both patient harm and financial costs. In order for the MDSS to be able to calculate those beliefs, it has to use a knowledgebase with clinical knowledge. The clinical knowledge contains information about the observations (such as symptoms) given a certain disorder. Currently this knowledge is manually entered by clinicians into a semantic knowledgebase. This process is very costly in terms of human effort, which makes the automation of this process a very interresting topic. In this article we will examine two approaches to generate Bayesian knowledge from real life data. Important limiting factors are the computationability of the generated models in real time decision support and the ability to easily validate the generated knowledge with the results of clinical trials, hence the transparency of the model.

At first, naive Bayesian networks are used to tackle this problem. Later, in search of more specific models premises are used, e.g. models per gender can be generated. The possibility of logical programming will be examined to find relevant premises.

As a use case the *Heart Disease Dataset* from the UCI *Machine Learning Repository* is used [1].

2 Objectives and approach

The main objective is to find clinically relevant knowledge about observations, given a certain disorder. Eventhough the inverse relation is used for diagnosis, the knowledge is saved in this way to avoid context dependence in the data. e.g. in the case of a flu epidemic most patients who have fever will be diagnosed having flu, but when the epidemic is over, fever will typically have a different cause. The chances that a patient with flu shows fever as a symptom, however, stayed (more or less) the same. To be usable in a real world MDSS, the knowledge has to be modeled in such a way that the inverse relation can be calculated in real time. Other requirements are transparency to facilitate the validation of the results with the knowledge from clinical trials.

Two approaches will be used. Firstly naive

Bayesian networks are used to tackle this problem. Later the ILP-framework Toplog [3],[4] is used to find valuable premises which outline more specific naive Bayesian networks. Toplog uses *top directed hypothis derivation* to define the hypothis search space.

Results from both approaches will be compared afterwards.

3 Related work

A common way to generate classifiers which perform even better than naive Bayes [6], yet remain a low combinatorial impact is to use *tree augmented naive Bayes* (TAN) (Friedman et al.[2]). While examining this possibility we found that contrary to the literature [2] the model was not stable, for 4 different subsets of the data set and the total data set, Five very different models were found by Weka [5] . More importantly, none of the found models had any clinical relevance.

4 Unconditional naive Bayesian networks

A naive Bayesian network consists of a class variable node, which is directly connected to several other nodes with two stoichastic parameters s and a.

sensitivity (s) The chance a disorder causes a certain observation

$$\begin{aligned} s &= P(test = true | actual = true) \\ &= \frac{TP}{TP + FN} \end{aligned}$$

aspecificity (a) The chance an observation is caused by anything but the disorder.

$$\begin{aligned} a &= 1 - P(test = false | actual = false) \\ &= 1 - \frac{TN}{TN + FP} \end{aligned}$$

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}} \quad (1)$$

The main challenge here is the binarization of ranges of observations into a risk and a non risk class. To do this we used a heuristic to find a good starting point from where a local search was done using the Mathews correlation coefficient (MCC).

```
x is the number of patients with a
disorder;
y is the total number of patients;
Sort patients by parameter;
if positive correlation between
parameter and disorder then
    cut-off value is parameter value of
    patient number x ;
end
else
    cut-off value is parameter value of
    patient number (y-x) ;
end
```

Algorithm 1: Binariastion Heuristic

5 Conditional naive Bayesian networks

Toplog uses the input vector $S_{TDHD} = \langle NT, \top, B, E \rangle$ where NT is a set of "non terminal"predicate symbols prefixed with a `$`-sign. $\top$ is a logic program which represents the declarative bias over the hypotheses space. B is the background knowledge, E are positive and negative examples. Every clause $\top_i \in \top$ has at least one element from NT while B and E cannot contain any element from NT.

The goal of TDHD is to find a consistent set of hypothesis clauses H, without elements from NT for which every element $h \in H$ there is at least one positive example $e \in E$ for which the following holds:

$$\top \models h \quad (2)$$

$$B, h \models e \quad (3)$$

Mode declarations are used in this thesis to define the hyposthese space. Toplog translates mode declarations automaticly to a $\top$-theory. `modeh` and `modeb` define the head and the body respectively from the hypothesis $h \in H$.

For example:

```
modeh(observation_i(+patient)).
modeb(observation_j(+patient)).
modeb(disorder_a(+patient)).
```

is translated to the $\top$-theory:

$\top_1$: `observation_i(X)` $\leftarrow$ `$body(X).`
$\top_2$: `$body(X)` $\leftarrow$ `. %an empty body`

$\top_3$: `$body(X)` ← `observation_j(X).`
$\top_4$: `$body(X)` ← `disorder_a(X).`

The set NT is defined by $NT = \$body$

The background knowledge B is a set of facts about patients:

```
observation_j(patient_x).
observation_j(patient_y).
disorder_a(patient_y).
...
```

At last, the positive and negative examples are added to the S_{TDHD}-vector, depending whether or not the observation from the head of the hypothis is observed, for every patient. For positive examples a positive value is given as a parameter, for negative examples a negative one. The absolute value indicates the weight for the example. In this case all patients weigh equally.

```
example(observation_i(patient_x),1).
example(observation_i(patient_y),-1).
...
```

5.1 Top Directed Hypothesis Derivation

The derivation of the hypotheses is done in stages. Firstly an example e is proven using the backgroundknowledge B and the top level theory $\top$. The $\top$-theory is executed using the random example e_1 as a starting clause. This execution yields the resolution which consists of a number of clauses from the $\top$-theory and the background knowledge B.

By using the previously described $\top$-theory and the following clause from the background knowledge b_1 = `observation_j(patient_x)`. The refutation of the negation of the example e_1 = `example(observation_i(patient_x),1)` yields the following refutations: $r_1 = \langle \neg e, \top_1, \top_2 \rangle$ and $r_2 = \langle \neg e, \top_1, \top_3, b_1, \top_2 \rangle$.

In a second stage the refutations R are reordered to the form $R' = D_h R_e$ where D_h is an SLD-resolution of the hypothis h for which (2) and (3) hold. A formal proof for this is given in [4]. The following clauses are found by applying SLD resolution to r_1 and r_2:

- h_1 =`observation_i(X)` with $d_{h1} = \langle \top_1, \top_2 \rangle$
- h_1 =`observation_i(X)` ← `observation_j(patient_x)` with $d_{h2} = \langle \top_1, \top_3, \top_2 \rangle$

5.2 Converting Toplog results to Bayesian rules

The naive Bayesian network uses rules like $P(observation_i|disorder_a)$, with a sensitivity s and an aspecificity a. The same rule can be written in Prolog as follows:

```
observation_i(X) :- disorder_a(X).
```

With the input file, described in the previous section, Toplog generates rules such as the following:

```
observation_i(X)
  :- disorder_a(X), observation_j(X).
```

In other words: `observation_i` will be observed if `disorder_x` *and* `observation_j` are observed. For example:

```
trestbps(X) :- heartdisease(X), age(X).
```

Patient X is a member of the risk class when it comes to heart beats per minute in rest (>= 132*bps*) *if* one or more vessels are more than 50% narrowed *and* the patient also belongs to the risk class for the observation "age"(55 years old or older). With sensitivity 0,74 and aspecificity 0,38.

The same rule can be rewritten as:

$$P(observation_i|disorder_a, observation_j)$$

or:

$$observation_j \Rightarrow P(observation_i|disorder_a)$$

when ***observation_j*** *is observed, then* $P(observation_i|disorder)$ *with sensitivity s and specificity a.*

Because the rules found by Toplog take several observations into account, they yield more specific results, which in general have a significantly higher sensitivity then the general Bayesian rules.

6 Evaluation

The resulst of both approaches can be found in the tables 1 and 2. The second approach yields an overall increased sensitivity and an (albeit less significant) increase of the aspecificity, compared to the general naive Bayes. For most observations a premise can be found which yields a higher MCC.

The increased aspecificity is a result of the way the background knowledge B is specified.

Only positive statements are allowed, in contrary to the examples where both positive and negative examples are allowed. This results in a bias to the upper right corner of the ROC-curve.

6.1 Validation

Both approaches delivered medically valid results whenever the data was realistic. E.g. the sensitivities found for high cholesterol levels and for an exercise induced angina were unrealisticly low. However, the unrealisticly low correlation between those observations and the presence of heart failure was also shown in the histograms of these parameters.

7 Conclusions and future work

Two new knowledgebases were genereated using the described approaches. A general naive Bayesian network which applies to all observations from the dataset and more specific Bayesian networks which depend on premises. Both results can be used by the MDSS, since it can handle several knowledge bases.

Because wrong diagnoses in medical application should be avoided, it is necessary that the data mining results are validated by a medical expert. Data mining can be used as a powerful tool, however, because of the above mentioned reasons it can never be done completely automated.

7.1 Heuristics

Alternative heuristics for the binarization could be found. Although it didn't appear in the use case, it is possible that the distribution of patients with a disorder cannot be approximated with a normal distribution. The used binarization will not be applicable in such cases, i.e when a disorder occurs both with young children and elderly. In such cases a heuristic which splits the observation in several classes and identifies the classes with a high risk for a disorder is needed.

A second challenge is to find a heuristic which is able to find a dependency of the threshold value in function of another observation. Such as an age related threshold for blood pressure. A high quantity of high quality data is needed in order to find such subtile correlations. For each age class, for example, a blood pressure threshold should be found with a relatively small error margin before one can conclude there is such a correlation. Evaluating all possible correlations between observations and thresholds is likely to cause a combinatorial explosion. To tackle this meta heuristics might be employed. Alternatively expert knowledge might be used to limit the search space, e.g. a medical might indicate there might be an interresting correlation between age and a blood pressure threshold. The feasibility of exhaustively searching this space is much more likely, than a much bigger search space with all posisble correlations.

8 Acknowledgements

Sofie Uvin, for this template.

References

[1] David W. Aha. UCI machine learning repository, heart disease databases, 1988.

[2] Nir Friedman, Dan Geiger, and Moises Goldszmidt. Bayesian network classifiers. *Mach. Learn.*, 29(2-3):131–163, 1997.

[3] Stephen Muggleton, José Carlos Almeida Santos, and Alireza Tamaddoni-Nezhad. Toplog: Ilp using a logic program declarative bias. In Maria Garcia de la Banda and Enrico Pontelli, editors, *ICLP*, volume 5366 of *Lecture Notes in Computer Science*, pages 687–692. Springer, 2008.

[4] José Carlos Almeida Santos. Toplog: Ilp using a logic program declarative bias. TRANSFER REPORT. 2008.

[5] Weka Machine Learning Project. Weka. URL http://www.cs.waikato.ac.nz/~ml/weka.

[6] Ian H. Witten and Eibe Frank. *Data Mining: Practical Machine Learning Tools and Techniques, Second Edition (Morgan Kaufmann Series in Data Management Systems)*. Morgan Kaufmann, June 2005.

CONDITION	SENSITIVITY	ASPECIFICITY	MCC
age >= 55	0.7	0.42	0.29
sex = 1	0.64	0.26	0.31
cp = 1	0.44	0.56	-0.06
cp = 2	0.14	0.66	-0.41
cp = 3	0.36	0.61	-0.21
cp = 4	0.8	0.28	0.52
trestbps >= 132	0.62	0.5	0.12
chol >= 241	0.55	0.43	0.13
fbs = 1	0.69	0.49	0.15
restecg = 0	0.52	0.62	-0.1
restecg = 1	0.66	0.53	0.11
restecg = 2	0.57	0.55	0.02
thalach <142	0.71	0.37	0.35
exang = 1	0.84	0.37	0.47
oldpeak >= 0.889	0.77	0.37	0.4
slope = 1	0.39	0.78	-0.39
slope = 2	0.78	0.48	0.31
slope = 3	0.78	0.63	0.1
ca >= 1	0.75	0.27	0.48
thal = 3	0.3	0.8	-0.51
thal = 6	0.77	0.55	0.14
thal = 7	0.81	0.39	0.42

Table 1: Results for naive Bayes

CONDITION	PREMISSES	S	A	MCC
age >= 55	restecg>=1	0.74	0.39	0.36
age >= 55	sex=1	0.76	0.51	0.26
age >= 55	thal=3	0.45	0.18	0.3
trestbps >= 132	age>=55	0.73	0.66	0.09
chol >= 241	slope =2 AND exang = 1	0.91	0.77	0.19
chol >= 241	slope =2	0.79	0.64	0.17
chol >= 241	exang = 1	0.85	0.76	0.12
thalach <142	sex=1	0.78	0.43	0.35
thalach <142	exang=1	0.88	0.71	0.21
oldpeak >= 0.89	exang=1	0.89	0.72	0.2
ca >= 1	age>=55	0.81	0.33	0.49
ca >= 1	chol>=241 AND exang=1	0.92	0.72	0.27
ca >= 1	sex=1	0.85	0.33	0.52
sex=1	cp=4	0.83	0.56	0.24
sex=1	age>=55	0.76	0.43	0.28
sex=1	ca>=1	0.85	0.47	0.38
exang=1	thalach <142 AND slope =2	0.9	0.77	0.19
exang=1	thalach <142 AND oldpeak >=0.89	0.9	0.71	0.24
exang=1	cp=4 AND oldpeak >=0.89	0.93	0.77	0.23
cp=4	sex=1	0.83	0.36	0.49
cp=4	oldpeak>=0.89	0.89	0.45	0.47
restecg=1	age >=55	0.8	0.67	0.13
slope=2	oldpeak>=0.89	0.82	0.66	0.18
slope=2	sex=1 AND age>=55	0.85	0.75	0.13
slope=2	cp=3	0.51	0.2	0.34
thal=7	sex=1	0.81	0.49	0.33
thal=7	oldpeak>=0.89	0.91	0.47	0.49

Table 2: Results for naive Bayes with premisses

Bijlage G

Poster

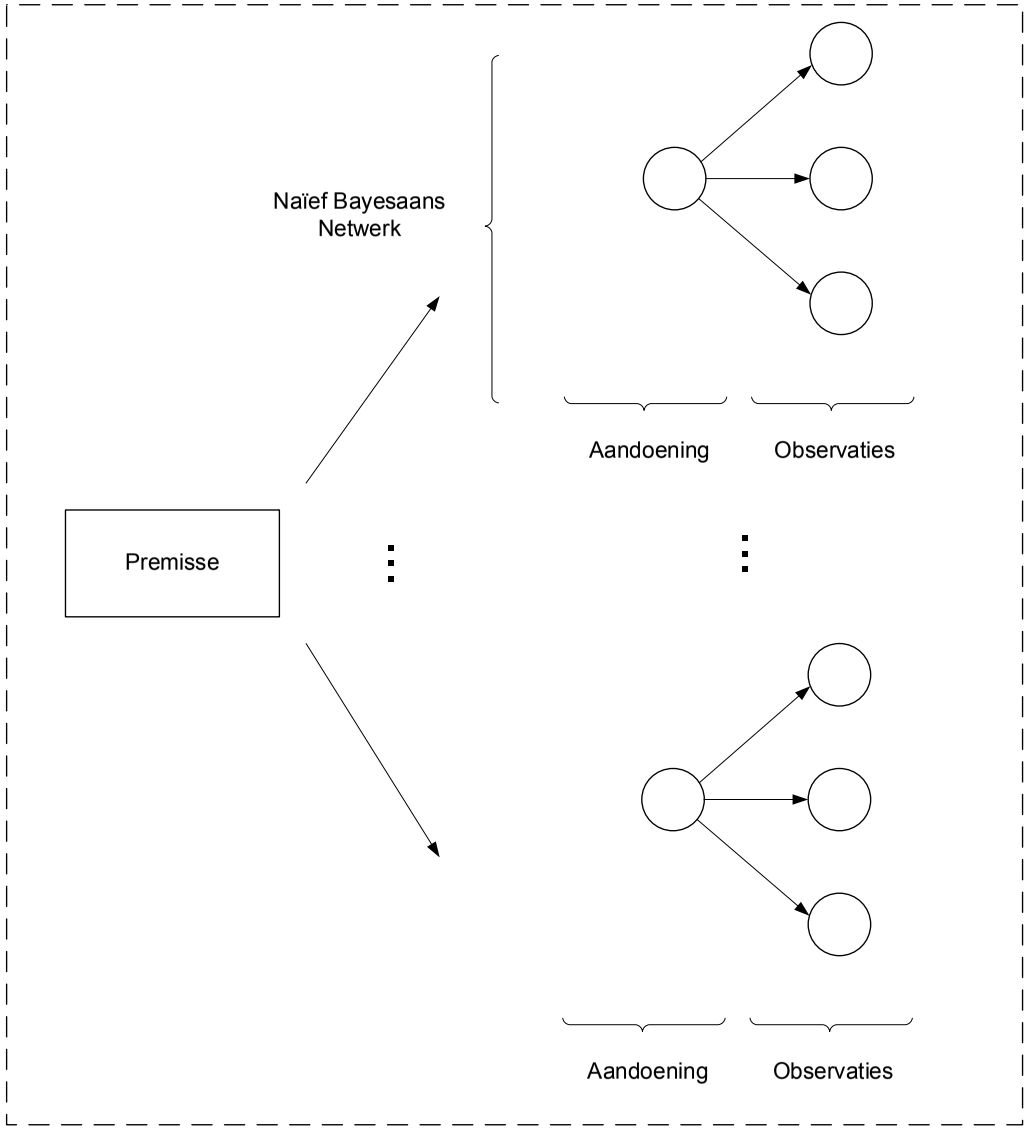
Genereren van Medische Kennis uit Praktijkdata

Promotor Agfa HealthCare:
ir. Jos De Roo
Promotor KaHo Sint-Lieven:
dr. ir. Annemie Vorstermans
Co-promotor KaHo Sint-Lieven:
lic. ing. Joris Maervoet
Student:
Boris De Vloed

Praktijkdata

De databank met praktijkdata bevat per patiënt:

- Observaties zoals:
  - leeftijd, geslacht, …
  - resultaten van tests zoals bloedonderzoek ed.
- Diagnose:
  Het al dan niet vaststellen van een bepaalde aandoening



- Uit een databank met observaties van patiënten en diagnoses wordt op een geautomatiseerde manier kennis gegenereerd over de observatie die gepaard gaan met een bepaalde aandoening.
- Bij continue waarden zoals leeftijd, cholesterolgehalte, etc. wordt een optimale waarde gevonden om de observaties op te splitsen.
- Er kunnen meer specifieke Bayesaanse Netwerken berekend worden door een premisse mee te geven. Zoals bvb.: verschillende modellen per geslacht.
- De resultaten kunnen geëxporteerd worden als regels in Notation3. Deze kennis kan dan als kennisbron gebruikt worden voor een semantische reasoner.
- Met een gemodificeerde versie van het Prolog programma Toplog wordt onderlinge afhankelijkheid van de observaties nagegaan.

Bayesaans Netwerk
De link tussen aandoening en observatie bevat 2 waarden:

- Sensitiviteit (s):
  De kans dat een aandoening een bepaald gevolg veroorzaakt.
- Aspecificiteit (a):
  De kans dat alles behalve de aandoening een bepaald gevolg veroorzaakt.

$$s = P(Observatie == True \mid Aandoening == True) = \frac{True\ Positives}{True\ Positives + False\ Negatives}$$

$$a = 1 - P(Observatie == False \mid Aandoening == False) = 1 - \frac{True\ Negatives}{True\ Negatives + False\ Positives}$$

Medische Praktijkdata → Medische Kennis