

Antwoorden Tentamen Kennissystemen

24 maart 2004

13:30 – 16:30 uur

Puntenverdeling:

1a	1b	2a	2b	3a	3b	4a	4b	5a	5b
5	5	10	10	5	10	15	10	10	10

Tentamencijfer = (puntentotaal + 10) / 10

Opgave 1: Basisbegrippen

- a) Wat zegt de *physical symbol hypothesis* en waarom is deze hypothese van belang voor kennissystemen?

Letterlijk: "A physical symbol system has the necessary and sufficient means for general intelligent action."

In ieder geval beschrijven dat symbolen een noodzakelijke voorwaarde en een voldoende voorwaarde voor intelligent gedrag is.

Deze hypothese is van belang omdat kennissystemen symboolsystemen zijn. We kunnen er alleen vanuit gaan dat kennissystemen intelligente dingen kunnen doen als we deze hypothese voor waar aannemen.

- b) Wat is de basisstructuur van een kennissysteem en wat zijn de voordelen van deze structuur?

- *Inference engine*
- *Knowledge base*
- *(User interface)*

Voordeel: mogelijk hergebruik van redeneermachine en kennisbank.

Opgave 2: Tijd en ruimte

- a) Noem vier (en als dat niet lukt zo veel mogelijk) verschillende keuzes die je kunt maken in het modelleren van tijd, en geef bij iedere keuze twee voorbeelden.

Vier van de volgende:

- *discrete of continue tijd representatie (m.b.v. tijdspunten vs. lijn)*
- *granulariteit (kleinste onderscheidbare eenheid, vb. seconden vs. dagen, etc)*
- *kwantitatieve of kwalitatieve representatie (exact vs. eerder / later)*
- *gebeurtenissen registreren op punten of intervals*
- *discrete of continue waarden (op punt x is iets hoog / laag vs op punt x is iets exact...)*
- *waarden fixeren per punt of per interval (d.w.z., als waarden uitgezet worden tegen de tijd resulteert dit in een respectievelijk een lijn-diagram of een staafdiagram)*
- *een linear, parallel, of branching tijds-model (één tijdslijn, meerdere parallelle tijdslijnen, of een tijdslijn die splitst)*

- b) Wat is het voordeel van een *quad-tree* representatie van ruimte? Beschrijf wat deze representatie inhoudt.

De quad-tree representatie maakt het mogelijk efficiënt te zoeken in niet uniform verdeelde ruimtes. Deze representatie verdeelt de ruimte in delen met een ongeveer gelijke hoeveelheid elementen, door eerst de ruimte allereerst in vier kwadranten te verdelen en vervolgens ieder kwadrant dat meer elementen bevat dan een bepaalde drempelwaarde opnieuw in kwadranten te verdelen, dit net zo lang herhalend totdat ieder kwadrant minder elementen dan de drempelwaarde bevat.

Opgave 3: Bayes

- a) Waarom is de regel van Bayes vaak nuttig?

Omdat we vaak weten hoe vaak symptoom X gegeven Y voorkomt, terwijl we geïnteresseerd zijn in de kans op Y gegeven symptoom X.

- b) Stel dat we weten dat de kans op koorts 1% is. We weten verder dat 90% van de mensen met koorts rode vlekken heeft. We nemen aan dat deze kansen gelijk zijn voor mensen die een tentamen doen en mensen die dat niet doen. We weten ook dat 50% van de mensen die tentamen doen rode vlekken hebben en dat slechts 5% van de mensen die geen tentamen doen rode vlekken hebben.

Je kijkt tijdens het tentamen naast je en ziet dat je buurman rode vlekken heeft. Hoe groot is de kans dat 'ie koorts heeft? Laat je berekening zien.

*Gevraagd: kans op koorts gegeven rode vlekken $P(K|R)$. We weten de kans op rode vlekken gegeven koorts $P(R|K) = 0.9$. We weten ook dat kans op koorts $P(K) = 0.01$. Bayes' regel zegt dat $P(K|R) = P(R|K) * P(K) / P(R)$. Tijdens een tentamen is de kans op rode vlekken $P(R) = 0.5$. De kans op koorts is dan $P(K|R) = 0.9 * 0.01 / 0.5 = 0.009 * 2 = 0.018 = 1,8 \%$.*

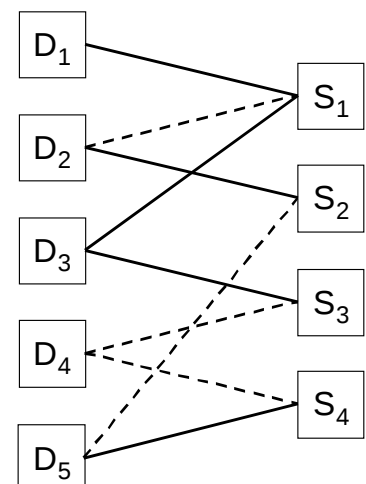
Je zit in de bus en ziet iemand met rode vlekken. Hoe groot is de kans dat 'ie koorts heeft? Laat je berekening zien.

*Zelfde argumentatie. Enige verschil is dat kans op rode vlekken zonder tentamen $P(R) = 0.05$. De kans op koorts is dan $P(K|R) = 0.9 * 0.01 / 0.05 = 0.009 * 20 = 0.18 = 18 \%$.*

Opgave 4: Classificatie

- a) Ga uit van het classificatie model dat hiernaast staat gegeven. Beantwoord de volgende vragen:

- Geef een datavector die *geen* oplossingen geeft (zonder gebruik te maken van onbekende waarden "?").



*Algemene uitleg: een vector specificeert de data waarden D_i . Een doorlopende lijn naar oplossing S_i betekent dat D_i de waarde 1 moet hebben voor die oplossing, een onderbroken lijn naar S_i vereist waarde 0. Oplossing S_1 vereist dus $\langle 1,0,1,?,? \rangle$
Antwoord op vraag: een datavector zonder oplossing is bijvoorbeeld $\langle 0,0,0,1,0 \rangle$. Er zijn er vele andere goede antwoorden mogelijk.*

- Geef een datavector die tot een samengestelde oplossing leidt.

Een vector waarvoor meerdere oplossingen nodig zijn om alle data te verklaren, bijvoorbeeld $\langle 1,0,1,0,0 \rangle$, deze vereist zowel S_1 als S_3 .

- Geef voor de datavector $\langle 1,1,?,1,0 \rangle$ alle consistente, inconsistente en matching oplossingen aan.

Inconsistent = minimaal één vereiste waarde is bekend maar niet correct. Dus S_1 (vanwege D_2), S_3 (vanwege D_4) en S_4 (vanwege D_4 en D_5).

Consistent = niet inconsistent, dus S_2 .

Matching = alle vereiste waarde zijn bekend en correct, dus wederom S_2 .

- b) Leg uit waarin de classificatiemethode MC4 (*data-driven hierarchical classification*) verschilt van MC3 (*solution-driven hierarchical classification*). Leg meer uit dan dat de een *data-driven* is en de ander *solution-driven* ;-)

Beide zijn hiërarchische methoden. MC4 gebruikt een aantal initiële observaties om een aantal seed-classes te bepalen en doet daarna verdere observaties om de precieze class te bepalen. MC3 doet slechts een bread-first search door de oplossingen ruimte.

Opgave 5: Configuratie

- a) Beschrijf kort wat het drempel-effect (*threshold effect*) is dat kan optreden bij configuratie en geef een voorbeeld. Wat is een vaak gebruikte oplossing voor dit probleem?

Een kleine verandering in de requirements kan een grote verandering in de oplossing met zich meebrengen.

Voorbeeld voorbeeld: CD's opbergen in je CD rek: vijftig passen in één rek, voor 51 CD's heb je al twee rekken nodig...

Veel gebruikte oplossing is om de requirements enigszins te overschatten.

- b) Leg uit wat de *key-component approach* is.

*Er is een functionele beschrijving van een configuratie. Voor iedere functie is er een bepaalde "standaard component" (*key-component*) gedefinieerd. Bij een configuratie-taak wordt een initiële configuratie gegenereerd door de standaard componenten te kiezen die voor de gespecificeerde functies bestaan. Vervolgens worden componenten die vereist zijn door de *key-components* toegevoegd en een arrangement gemaakt.*

Einde tentamen.