

Oefententamen IN2205 Kennissystemen

20 December 2007

- Tijdens een echt tentamen is gebruik van boek of aantekeningen niet toegestaan.
- De lengte van dit tentamen is niet representatief voor het echte tentamen, het type vragen wel.
- Uiteraard komen in één tentamen niet alle onderwerpen aan bod. Trek daarom op basis van dit tentamen geen conclusies over stof die nooit getoetst wordt.
- Wat betreft de meerkeuzevragen:
 - Er is voor iedere vraag telkens maar één goed antwoord mogelijk.
- Wat betreft de open vragen:
 - Geef antwoord in correct Nederlands of Engels en schrijf leesbaar (gebruik eerst kladpapier).
 - Geef geen irrelevante informatie. Dit kan leiden tot puntenaftrek.
- Voordat je je antwoorden inlevert, controleer of op ieder blaadje je naam en studienummer staat en geef het aantal ingeleverde bladen aan op (tenminste) de eerste pagina.
- De tentamenstof bestaat uit Chapters 1,2,3,4,6,7,8 en 9 uit Artificial Intelligence van Michael Negnevitsky, behalve secties 3.4, 3.5, 8.2, 9.1 en 9.7. Daarnaast behoren het artikel *Swarm Smarts* van Eric Bonabeau et. al. en de lecture notes over *Bayesian Reasoning en Swarm Intelligence* tot de stof.
- Totaal aantal pagina's: 10.

Multiple-choicevragen

1. Waarvoor gebruiken expert systemen conflict resolution?
 - A. Om te bepalen of forward chaining of backward chaining gebruikt moet worden.
 - B. Om te bepalen welke consequent de grootste kans heeft.
 - C. Om te bepalen welke regel mag vuren als meer dan één regel kan vuren.
 - D. Om te bepalen welke regels tot een bepaalde conclusie hebben geleid.

Antwoord: C.

Om te bepalen welke regel mag vuren als meer dan een regel kan vuren.

2. In een regelgebaseerd kennissysteem kunnen we kansrekening of *certainty factors* gebruiken om onzekerheid te modelleren. Welke uitspraak klopt?
 - A. Certainty factors bieden mogelijkheid om een oorzaak voor een feit weg te redeneren als een andere mogelijke oorzaak waar is.
 - B. Certainty factors zijn waarden tussen 0 en 1.
 - C. Allebei.
 - D. Geen van beide.

Antwoord: D. Geen van beide.

3. Fuzzy logic wordt gebruikt om te redeneren met ...
 - A. onzekere informatie
 - B. onvolledige informatie
 - C. onprecieze informatie
 - D. onbegrijpelijke informatie

Antwoord: C. onprecieze (onnauwkeurige / vage) informatie.

4. Stel we hebben de fuzzy sets *prijzig* en *goedkoop* met membershipfuncties:

$$\mu_{prijzig}(x) = (0/1, 0/5, 0.1/10, 0.25/50, 0.5/100, 0.6/125, 0.7/150, 0.8/200, 1/250) \quad (1)$$

$$\mu_{goedkoop}(x) = (1/1, 1/5, 0.7/10, 0.6/50, 0.3/100, 0.2/125, 0.1/150, 0/200, 0/250) \quad (2)$$

Wat is de membershipfunctie van de fuzzy set *prijzig of niet goedkoop* als we gebruik maken van de *probor* en *prod* operatoren?

- A. $\mu(x) = (0/1, 0/5, 0.1/10, 0.25/50, 0.5/100, 0.6/125, 0.7/150, 0.8/200, 1/250)$.
- B. $\mu(x) = (0/1, 0/5, 0.37/10, 0.55/50, 0.85/100, 0.92/125, 0.97/150, 1/200, 1/250)$.
- C. $\mu(x) = (0/1, 0/5, 0.03/10, 0.1/50, 0.35/100, 0.48/125, 0.63/150, 0.8/200, 1/250)$.
- D. $\mu(x) = (1/1, 1/5, 0.8/10, 0.75/50, 0.5/100, 0.4/125, 0.3/150, 0.2/200, 0/250)$.

Antwoord: B. $\mu(x) = (0/1, 0/5, 0.37/10, 0.55/50, 0.85/100, 0.92/125, 0.97/150, 1/200, 1/250)$

5. Bijwoorden als *erg (very)*, *extreem*, *min of meer (more or less)*, *enigszins* enz. kunnen linguïstische waarden afzwakken of versterken. Indien de linguïstische waarde *gezellig* een fuzzy set is met membership functie $\mu(x)$, dan kan de fuzzy set *erg gezellig* worden gekarakteriseerd door de membership functie $[\mu(x)]^p$. Voor p geldt:
- A. $p = 0.5$
 - B. $p = 2$
 - C. $p = 0.3$
 - D. $p = 3$

Antwoord: B.

$p = 2$ zie boek bladzijde 97.

6. Welke soort learning is bij het Kohonen netwerk aan de orde?
- A. Unsupervised learning.
 - B. Supervised learning.
 - C. Reinforcement learning.
 - D. Hebb learning.

Antwoord: A.

Unsupervised learning.

7. We willen een systeem ontwikkelen dat een niet technische gebruiker kan ondersteunen bij het installeren en onderhouden van een operating system. We kunnen het beste gebruik maken van ...
- A. neurale netwerken.
 - B. expert systemen.
 - C. fuzzy logic.
 - D. swarm intelligence.

Antwoord: B. expert systemen

8. We willen een camera ontwikkelen die zelf kan vaststellen of er gevaar dreigt in een openbare ruimte. We kunnen het beste gebruik maken van ...
- A. neurale netwerken.
 - B. expert systemen.
 - C. fuzzy logic.
 - D. swarm intelligence.

Antwoord: A. neurale netwerken

Open vragen

9. Een grote supermarktketen heeft winkels verspreid over het hele land. Deze winkels worden dagelijks bevoorrad door een vloot van vrachtwagens die vertrekken vanuit een centraal gelegen bevoorradingscentrum. Een vrachtwagen kan per rit genoeg lading meenemen om drie supermarkten te bevoorraden.

- (a) (1 punt) De supermarktketen heeft behoefte aan een systeem dat de ritten zou plant dat het totale aantal kilometers gemaakt door de vrachtwagens zo klein mogelijk is. Welke van de in het vak behandelde technieken zou jij adviseren om dit systeem met te implementeren en waarom?

Antwoord: Dit zou je met behulp van ant-based algorithms kunnen doen. Een 'mier' moet dan een route langs alle supermarkten vinden waarbij hij na het bezoeken van maximaal drie supermarkten weer terug moet keren naar het bevoorradingscentrum.

Merk op dat we alleen het totale aantal kilometers hoeven te minimaliseren (en bijvoorbeeld niet de hoeveelheid tijd) het aantal vrachtwagens is dus niet direct van belang: het maakt dus niet uit of we alle ritten door een vrachtwagen laten uitvoeren of iedere rit door een andere vrachtwagen.

Als alternatief zouden genetische algoritmen gebruikt kunnen worden. Een chromosoom stelt de combinatie van alle ritten voor (we zouden kunnen volstaan met de lijst van supermarkten en bij het decoderen van het genotype de afstanden van en naar het bevoorradingscentrum kunnen toevoegen). De fitnessfunctie meet de totale lengte van een pad.

- (b) (1 punt) De supermarktketen opent een aantal nieuwe bevoorradingscentra. Niet ieder bevoorradingscentrum heeft ieder product op voorraad. De supermarkten geven dagelijks aan welke producten moeten worden aangevuld, dit kan per supermarkt verschillen zodat het aantal supermarkten dat een vrachtwagen kan bevoorraden varieert. Welke van de in het vak behandelde technieken zou je in dit geval aanraden om een planningssysteem te implementeren?

Antwoord:

We moeten hier een systeem ontwikkelen dat een planning oplevert die aan een aantal voorwaarden voldoet: de wensen van de supermarkten spelen een rol, de voorraad van de bevoorradingscentra, het aantal vrachtwagens en de capaciteit van de vrachtwagens. Genetische algoritmen zijn geschikt voor het implementeren van planningssystemen.

10. Stel, we hebben een knowledge base met de volgende regels:

1. IF p AND h THEN m
2. IF d AND b THEN c
3. IF s THEN q
4. IF r THEN t
5. IF c AND a THEN v AND f
6. IF g OR f THEN u
7. IF a THEN b
8. IF t THEN d
9. IF v THEN h
10. IF d AND q THEN a

In de database komen de volgende feiten voor: r, s.

- (a) Laat zien hoe de inference engine alle feiten afleidt die kunnen worden afgeleid. Geef per *cycle* aan welke regels vuren en hoe de database verandert. Ga uit van een seriële (depth-first) verwerking van regels.

Antwoord:

Regel vuurt	database
3	q , r, s
4	q, r, s, t
8	d , q, r, s, t
10	a , d, q, r, s, t
7	a, b , d, q, r, s, t
2	a, b, c , d, q, r, s, t
5	a, b, c, d, f , q, r, s, t, v
6	a, b, c, d, f, q, r, s, t, u , v
9	a, b, c, d, f, h , q, r, s, t, u, v

Meer regels kunnen niet vuren.

- (b) Laat zien hoe de inference engine aantoonst of doel *c* waar is. Geef aan welke feiten op de stack worden geplaatst en hoe de database verandert.

Antwoord: Database: r, s.

Doelen: c.

Stack: 2. Doelen c, b, d.

(We proberen doel d te vervullen. We zouden ook eerst doel b kunnen bekijken).

Stack: 2, 8. Doelen: c, b, d, t.

Stack: 2, 8, 4. Doelen: c, b, d, t, r.

Regel 4 vuurt, database: r, s, **t**.

Stack: 2, 8. Doelen c, b d.

Regel 8 vuurt, database: **d**, r, s, t.

Stack: 2. Doelen: c, b.

Stack: 2, 7. Doelen: c, b, a.

Stack: 2, 7, 10. Doelen: c, b, a, q. (d staat al in de database)

Stack: 2, 7, 10, 3. Doelen: c, b, a, q, s.

Regel 3 vuurt, database: d, r, **q**, s, t.

Stack: 2, 7, 10. Doelen: c, b, a.

Regel 10 vuurt, database: **a**, d, r, q, s, t. Stack: 2, 7. Doelen: c, b.

Regel 7 vuurt, database: a, **b**, d, r, q, s, t. Stack: 2. Doelen: c.

Regel 2 vuurt, database: a, b, **c**, d, r, q, s, t. Doel vervuld.

- (c) Benoem de methoden die je in (a) en (b) hebt gebruikt. Waarom heb je in (b) deze en niet een andere methode gebruikt?

Antwoord: Bij (a) forward chaining (of data-driven reasoning) en bij (b) backward chaining (of goal-driven reasoning). Backward chaining is efficiënter omdat het alleen de feiten bekijkt die nodig zijn om het doel te bewijzen. Forward chaining kan ook allerlei regels vuren die niet relevant zijn voor het doel.

Let op: Het antwoord: 'Ik heb backward chaining gebruikt omdat een doel gegeven is.', is niet juist. Dit is slechts het herhalen van vraag (b). De vraag is waarom backward chaining handig is als een doel gegeven is.

11. Dhr. Holmes heeft nogal moeite met opstaan. Gemiddeld verslaapt hij zich op 3 van de 5 werkdagen. Als Holmes zich verslaapt, dan is hij in 60% van de gevallen te laat op zijn werk. Op de andere dagen is hij slechts in 10% van de gevallen te laat. Als Holmes zich verslaapt, dan heeft hij vaak geen tijd om te ontbijten, zodat hij dit achter zijn bureau doet. Dit komt op de helft van de dagen dat hij zich verslaapt voor, anders eet hij nooit achter zijn bureau. De langere nachtrust maakt Holmes wel wakkerder gedurende de dag. Normaal komt hij in 30% van de gevallen slaperig over, als hij zich verslapen heeft komt hij slechts in 10% van de gevallen slaperig over.

We definiëren de volgende stochastische variabelen:

V : Holmes heeft zich verslapen

L : Holmes is te laat op zijn werk

E : Holmes eet achter zijn bureau

S : Holmes komt slaperig over

Wat is de kans dat Holmes zich verslapen heeft als hij op tijd op zijn werk is, maar achter zijn bureau zit te eten en wakker overkomt? Laat je berekening zien.

Antwoord:

$$P(V) = \begin{array}{|c|c|} \hline V & \\ \hline false & 0.4 \\ \hline true & \frac{3}{5} = 0.6 \\ \hline \end{array} \quad P(L|V) = \begin{array}{|c|c|c|} \hline L & V & \\ \hline false & false & 0.9 \\ \hline true & false & 0.1 \\ \hline false & true & 0.4 \\ \hline true & true & 0.6 \\ \hline \end{array} \quad P(E|V) = \begin{array}{|c|c|c|} \hline E & V & \\ \hline false & false & 1.0 \\ \hline true & false & 0.0 \\ \hline false & true & 0.5 \\ \hline true & true & 0.5 \\ \hline \end{array} \quad (3)$$

$$P(S|V) = \begin{array}{|c|c|c|} \hline S & V & \\ \hline false & false & 0.7 \\ \hline true & false & 0.3 \\ \hline false & true & 0.9 \\ \hline true & true & 0.1 \\ \hline \end{array} \quad (4)$$

We zoeken $P(V = true | L = false, E = true, S = false)$, toepassen van de regel van Bayes' en onafhankelijkheid tussen de 'symptomen' geeft:

$$P(V|L, E, S) = \frac{P(L|V)P(E|V)P(S|V)P(V)}{\sum_V P(L|V)P(E|V)P(S|V)P(V)} = \alpha P(L|V)P(E|V)P(S|V)P(V) \quad (5)$$

$$\alpha P(V|L = false, E = true, S = false) = \begin{array}{|c|c|} \hline V & \\ \hline false & 0.9 \times 0.0 \times 0.7 \times 0.4 = 0.0 \\ \hline true & 0.4 \times 0.5 \times 0.9 \times 0.6 = 0.108 \\ \hline \end{array} \quad (6)$$

$$\alpha = 0.0 + 0.108 = 0.108 \quad (7)$$

$$P(V|L = false, E = true, S = false) = \begin{array}{|c|c|} \hline V & \\ \hline false & 0.0/0.108 = 0.0 \\ \hline true & 0.108/0.108 = 1.0 \\ \hline \end{array} \quad (8)$$

De gezochte kans is dus 1.0.

12. *Fuzzy sets* vormen een generalisatie van *crisp sets*. Operaties op fuzzy sets kunnen worden uitgedrukt met *membership functions*. Membership van verzamelingen A en B wordt uitgedrukt als respectievelijk $\mu_A(x)$ en $\mu_B(x)$.

(a) Druk de membership function van $\neg A$, oftewel $\mu_{\neg A}(x)$ uit als functie van $\mu_A(x)$.

Antwoord: (p. 98)

$$\mu_{\neg A}(x) = 1 - \mu_A(x)$$

Antwoorden met \neg als aritmetische operator zijn fout.

(b) De AND-operatie van A en B wordt soms gedefinieerd als

$$\mu_{A \cap B}(x) = \mu_A(x) \times \mu_B(x). \quad (9)$$

Hoe is dan de OR-operatie gedefinieerd? Geef een afleiding. Hint: je kunt gebruik maken van het antwoord op vraag (a).

Antwoord: We gebruiken De Morgan.

$$\begin{aligned} \mu_{A \cup B}(x) &= \mu_{\neg(\neg A \cap \neg B)}(x) \\ &= 1 - \mu_{\neg A \cap \neg B}(x) \\ &= 1 - \mu_{\neg A}(x) \times \mu_{\neg B}(x) \\ &= 1 - ((1 - \mu_A(x)) \times (1 - \mu_B(x))) \\ &= 1 - (1 - \mu_A(x) - \mu_B(x) + \mu_A(x) \times \mu_B(x)) \\ &= 1 - 1 + \mu_A(x) + \mu_B(x) - \mu_A(x) \times \mu_B(x) \\ &= \mu_A(x) + \mu_B(x) - \mu_A(x)\mu_B(x) \end{aligned}$$

13. Om een fuzzy uitkomst van een fuzzy kennissysteem om te zetten naar een *crisp* uitkomst kunnen we *Sugeno inference* gebruiken. Leg slechts in grafieken uit hoe dit werkt. Er hoeven geen eenheden langs de horizontale as te staan.

Antwoord: Ik verwacht in vier grafieken als op bladzijde 113 van het boek onder "Aggregation of rule consequents". In plaats van om alleen plaatjes zou ik ook om de uitleg van de stappen of om berekeningen kunnen vragen!

14. De leerregel van Hebb voor het aanpassen van gewichten in neurale netwerken luidt:

$$\Delta w_{ij}(p) = \alpha y_j(p) x_i(p). \quad (10)$$

(a) Leg in woorden uit hoe leren met de regel van Hebb werkt.

Antwoord: De mate waarin de verbinding tussen i en j wordt versterkt hangt af van de mate waarin i en j tegelijk geactiveerd worden. Als beide vaak tegelijk een hoge activatie hebben dan wordt de verbinding tussen deze neuronen sterk. De regel van Hebb leert dus associaties tussen neuronen. (Die idee is gebaseerd op biologische theorie en over leren en biedt een alternatief voor het backpropagation algoritme waarvan is aangetoond dat het geen biologische tegenhanger heeft). De regel van Hebb wordt onder andere gebruikt in Hopfield netwerken.

(b) Waarom wordt vaak een forgetting factor aan deze leerregel toegevoegd?

Antwoord: Een forgetting factor verlaagt een gewicht enigzins (proportioneel met het oude gewicht (dus grote gewichten worden meer verlaagd en proportioneel aan de activatie van j)). Dit voorkomt dat de gewichten alleen maar kunnen stijgen en op den duur allemaal naar ∞ gaan.

15. (a) Geef aan onder welke omstandigheden een neurale netwerk een goede oplossing voor een probleem is.

Antwoord:

- Mogelijke taken zijn voorspellen, classificatie, clustering.
- Neurale netwerken zijn heel algemene modellen, kunnen iedere functie leren.
- Binaire en continue data.
- Kennis van domein niet beschikbaar of erg complex.
- Er moeten (veel) voorbeelden beschikbaar zijn, voor feedforward netwerken moet voor ieder van deze voorbeelden de juiste uitkomst bekend zijn.
- Ontbreken van data, ruis in data is geen probleem.

(b) Wat zijn de nadelen van neurale netwerken?

Antwoord:

- Er moeten voldoende voorbeelden beschikbaar zijn. Voor supervised learning moet de gewenste output voor ieder input bekend zijn.
- Juiste netwerkstructuur onduidelijk.
- Gevaar van overtrainen.
- Gevoelig voor locale maxima.
- Een neurale netwerk is een black box. Het is niet duidelijk hoe een netwerk van inputs naar outputs komt.

16. Genetische Algoritmen en Genetic Programming zijn twee vormen van Evolutionary Computation?

(a) Waarin verschillen Genetische Algoritmen en Genetic Programming?

Antwoord: Het belangrijkste verschil zit in de codering van chromosomen. Genetische Algoritmen gebruiken bitstrings met een vaste lengte. Genetic Programming gebruikt S-expressions: bomen opgebouwd uit terminals (variabelen, constanten) en functies die programma's representeren. De lengte van programma's ligt niet vast.

(b) Hoe werkt de standaard cross-over operator bij Genetische Algoritmen?

Antwoord: Standaard cross-over werkt op twee ouder-chromosomen (cross-over wordt met een bepaalde kans, meestal 0.7, toegepast op deze ouders). De operator kiest een cross-over punt binnen deze chromosomen. De eerste nakomeling krijgt alle bits voor dit cross-over punt van de eerste ouder en alle bits achter het cross-over punt van de tweede ouder. Voor de tweede nakomeling is het precies andersom.

(c) Wat is het nut van de mutatie operator bij Genetische Algoritmen?

Antwoord: De mutatie operator voorkomt dat het zoekproces in een lokaal maximum blijft steken. Door mutaties kunnen we meer van de oplossingsruimte te zien krijgen (maar er is geen garantie dat we de beste oplossing vinden).

17. Beschrijf hoe je met behulp van local best Particle Swarm Optimization het maximum kunt bepalen van de functie $f(a, b, c) = 23a^3 + b^c \times a$. Behandel de volgende onderwerpen: Representatie van een particle, beginpopulatie, aanpassen van de positie van een particle, terminatie.

Antwoord:

Er vanuit gaande dat $f(a, b, c)$ een maximum heeft (dit heb ik niet gecontroleerd):

1. Een particle i is een vector \mathbf{x}_i bestaande uit de drie parameters x, y, z .
2. De beginpopulatie bestaat uit n willekeurig gegenereerde particles (volgens een uniforme verdeling om een goede spreiding over de zoekruimte te krijgen) met snelheid 0 in alle drie de dimensies.
3. Om zijn positie aan te passen berekend een particle eerst de fitness van zijn huidige oplossing. De fitnessfunctie is in dit geval $f(x, y, z)$. Als de fitness van de huidige positie hoger is dan hoogste fitness die het particle tot nog toe gevonden heeft dan wordt de huidige positie de nieuwe beste positie \mathbf{y}_i van dit particle. Uit de beste posities van alle particles in de neighborhood van particle i wordt ook de beste positie gevonden door de neighborhood $\hat{\mathbf{y}}_i$ bepaald.

Het particle past nu zijn positie aan met een snelheid \mathbf{v}_i :

$$\mathbf{x}_i(t+1) = \mathbf{x}_i(t) + \mathbf{v}_i(t+1) \quad (11)$$

Deze snelheid volgt uit (per dimensie):

$$v_{ij}(t+1) = v_{ij}(t) + c_1 r_{1j} [y_{ij} - x_{ij}] + c_2 r_{2j} [\hat{y}_{ij} - x_{ij}] \quad (12)$$

De snelheid hangt dus af van de vorige snelheid, de afstand en positie van het particle t.o.v. van zijn beste positie en zijn afstand en positie t.o.v. de beste positie van de neighborhood. c_1 en c_2 zijn constanten die de bijdragen van deze componenten wegen. r_{1j} en r_{2j} zijn random getallen tussen 0 en 1.

4. als alle particles hun positie geupdate hebben wordt opnieuw de fitness van alle particles bepaald, als de (beste) gevonden goed genoeg is of een van te voren bepaald aantal iteraties doorlopen is of alle snelheden vrijwel nul zijn zodat de posities toch niet meer veranderen dan stopt het algoritme anders passen de particles hun posities opnieuw aan.

18. Stel dat een optimalisatieprobleem meerdere oplossingen heeft, d.w.z. dat de fitnessruimte behorende bij het probleem meerdere globale maxima heeft zou je dan kiezen voor genetische algoritmen of voor particle swarm

optimization om dit probleem aan te pakken? Ga er bij je antwoord vanuit dat oplossingen als een bitstring gerepresenteerd kunnen worden (er bestaat een binaire variant van particle swarm optimization, deze werkt hetzelfde als de continue variant).

Antwoord:

Particle Swarm Optimization werkt in dit geval beter dan de standaardvariant van genetische algoritmen. Dit zit in het verschil tussen de updatefuncties. Stel je hebt twee chromosomen die beide een hoge fitness hebben, maar verschillende maxima voorstellen. Cross-over leidt dan tot een punt dat waarschijnlijk ergens tussen deze maxima in ligt en dus slechter is dan zijn ouders. Bij local best Particle Swarm Optimization communiceert een particle slechts met een beperkt aantal burens, dat betekent dat er meerdere swarms kunnen ontstaan rond de verschillende maxima.

Einde tentamenopgaven.

Controleer voor de zekerheid of je alle vragen hebt beantwoord. Het zouden er 18 moeten zijn.