

AI en Data mining

---

**Van AI tot Data mining**

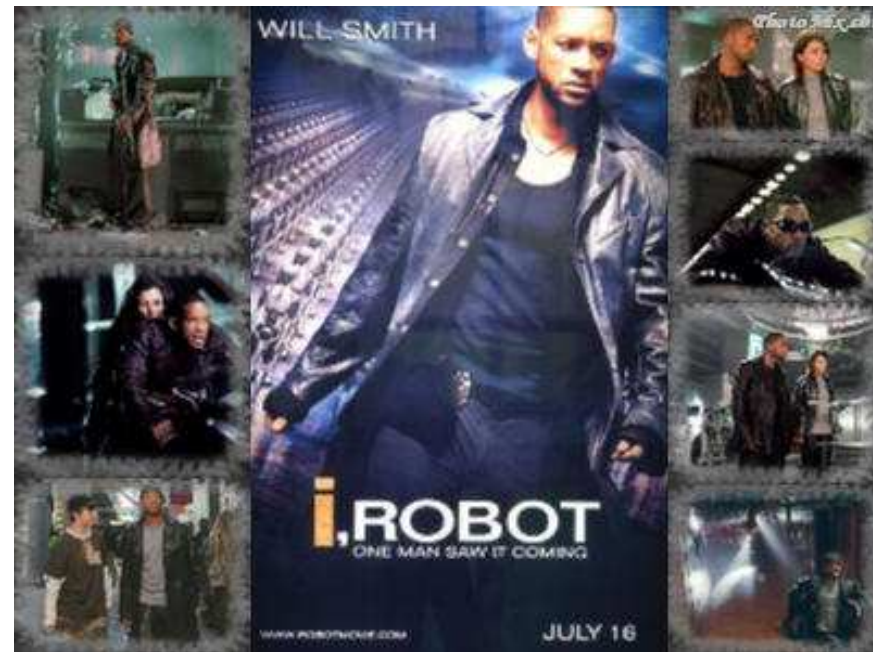
Van robots via boodschappenmandjes tot bio-informatica

Walter Kusters

Informatica, Universiteit Leiden

16 april 2007

[www.liacs.nl/home/kusters/](http://www.liacs.nl/home/kusters/)



**Kunstmatige intelligentie** (AI, Artificial Intelligence) is een verzamelnaam voor een heel breed vakgebied, met onder andere:

- *robotica*: Hoe programmeer je een robot?
- *data mining*: Hoe zoekt Google?
- *rechtspraak*: Word je volautomatisch be/veroordeeld?
- *vertalen*: “the spirit is willing but the flesh is weak” → ... → “the vodka is good but the meat is rotten”?
- *computer games*: Hoe speelt de computer bij Quake?
- *neurale netwerken*: Kun je beurskoersen voorspellen?

Je kunt op minstens **twee** manieren naar Kunstmatige intelligentie kijken:

1. vanuit een meer *psychologische* of *filosofische* richting:  
Wat is het verschil tussen een mens en een computer?  
Kan een computer denken?
2. vanuit een meer *technische* richting:  
Hoe werkt een schaakprogramma?  
Hoe werkt een Marsrobot?

Kunstmatige intelligentie laat computers zich zo gedragen dat het **intelligent** zou heten als mensen het op die manier zouden doen.

De beroemde **Turing-test** (1950) stelt het volgende probleem.

In een afgesloten kamer bevindt zich een mens *of* een computer, waarmee we alleen via een computerverbinding contact kunnen hebben. Kunnen wij bepalen of het een mens is, of juist een computer?

Het originele probleem was overigens met man  $\leftrightarrow$  vrouw.

In 1964–66 schreef Joseph Weizenbaum van het MIT in Boston het beroemde programma **ELIZA/DOCTOR**, dat een psychotherapeut “speelt” — met veel succes.

Een voorbeeldconversatie (Jason Hutchens’ HAL/HeX):

my name is bob

Pleased to meet you.

how are you doing, jas?

I am just fine, thanks! And you?

i am feeling great!

If you are feeling great, it’s your business, mate!

Dit soort computerprogramma’s doet mee in de jaarlijkse **Loebner-contest**.

Een **robot** is een “actieve, kunstmatige agent wiens omgeving de fysieke wereld” is. Het woord stamt uit 1921 (of eerder), en is gemaakt door de Tsjechische broers Capek. En **softbots**: RoboCom, internet programma’s.

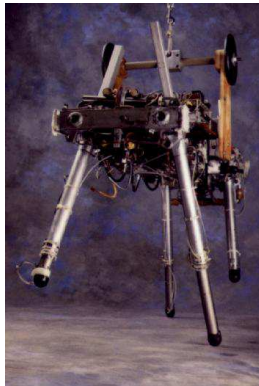
Van de science fiction schrijver Isaac Asimov (auteur van “I, Robot”) zijn de drie wetten van de **robotica**:

1. Een robot mag een mens geen kwaad doen.
2. Een robot moet menselijke orders gehoorzamen (tenzij dat tegen 1. ingaat).
3. Een robot moet zichzelf beschermen (tenzij dat tegen 1. of 2. ingaat).



NASA Sojourner

Sony Aibo



Marc Raibert's quadruped

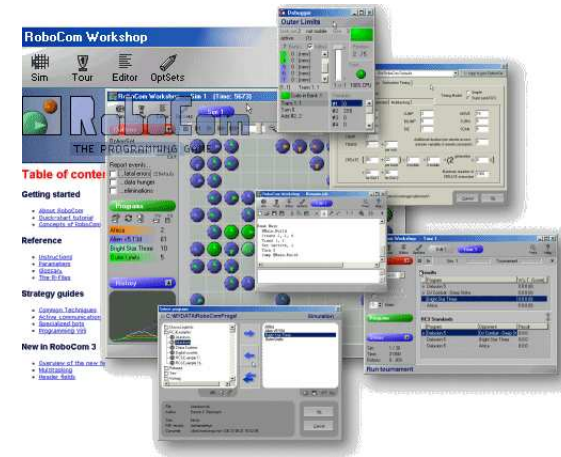
Honda



In het programma **RoboCom** vechten robot-programma's met elkaar.

; een klein robot-programma  
NAME Flooder/Shielder

```
BANK Een           ; eerste bank
  @Loop            ; label
  TURN 0           ; draai linksom
  SCAN #5          ; scan reference field
  COMP #5,0        ; leeg?
  JUMP @Loop       ; nee, verder draaien
  CREATE 2,1,0     ; ja; creeer nieuwe robot
  TRANS 1,1        ; en kopieer jezelf
  SET %Active,1    ; activeer hem/haar
  ; auto-reboot
```



<http://www.cyty.com/robocom/>

**Maxi** en **Mini** spelen het volgende eenvoudige spel: **Maxi** wijst eerst een horizontale rij aan, en daarna kiest **Mini** een verticale kolom.

3	12	8
2	4	6
14	5	2

Bijvoorbeeld: **Maxi** kiest rij 3, daarna kiest **Mini** kolom 2; dat levert einduitslag 5.

**Maxi** wil graag een zo groot mogelijk getal, **Mini** juist een zo klein mogelijk getal.

Hoe spelen we dit spel zo goed mogelijk?

Als **Maxi** rij 1 kiest, kiest **Mini** kolom 1 (levert 3); als **Maxi** rij 2 kiest, kiest **Mini** kolom 1 (levert 2); als **Maxi** rij 3 kiest, kiest **Mini** kolom 3 (levert 2). Dus kiest **Maxi** rij 1!

3	12	8
2	?	?
14	5	2

Nu merken we op dat de analyse hetzelfde verloopt als we niet eens weten wat onder de twee vraagtekens zit. Het  $\alpha$ - $\beta$ -algoritme onthoudt als het ware de beste en slechtste mogelijkheden, en kijkt niet verder als dat toch nergens meer toe kan leiden.

Ieder schaakprogramma gebruikt deze methode.



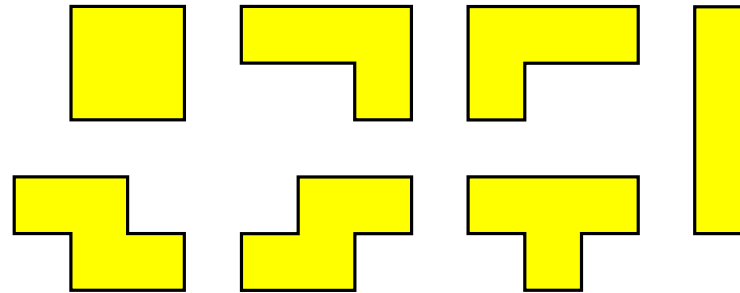
Ook aan een spel als **Tetris** kleven allerlei vragen:

- Hoe speel je het zo goed mogelijk? (AI)
- Hoe moeilijk is het? (complexiteit)
- Wat kan er allemaal gebeuren?

Zo is bijvoorbeeld bewezen dat sommige Tetris-problemen **NP-volledig** zijn, dat je bijna alle configuraties kunt bereiken, maar dat niet alle problemen “beslisbaar” zijn, zie:

[www.liacs.nl/home/kosters/tetris/](http://www.liacs.nl/home/kosters/tetris/)

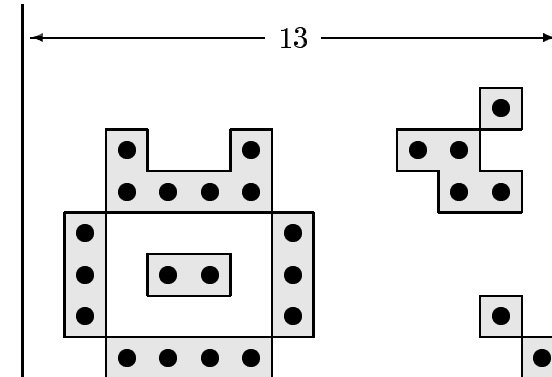
De 7 Tetris-stukken:



Stukken vallen random; volle regels worden verwijderd. De vraag “Kun je met een gegeven serie (inclusief volgorde) van deze stukken een bord helemaal leeg spelen?” is NP-volledig. (Ander voorbeeld: kun je een gegeven postzegelwaarde plakken?)

Als iemand het bord leeg speelt kun je dat eenvoudig controleren. Als het *niet* kan, kan men (tot nu toe) niks beters verzinnen dan alle mogelijkheden één voor één na te gaan!

Een “willekeurige” configuratie:



Deze kan gemaakt worden door 276 *geschikte* Tetris-stukken op de juiste plaats te laten vallen.

Let op: alleen geheel gevulde regels verdwijnen, alles daarboven zakt *één rij*.

Claim: op een bord van oneven breedte kan elke configuratie bereikt worden!

**Data mining** probeert interessante en (on)verwachte patronen te vinden in **grote** hoeveelheden (on)geordende data.

Bijvoorbeeld:

- Boodschappenmandjes: wat en hoe kopen we?
- Bio-informatica: DNA?

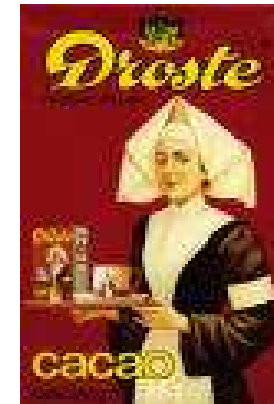
Problemen (selectie):

- resultaten: zowel verwacht als onverwacht
- bewegende doelen: steeds andere eisen
- data vergaren: wat/hoeveel kan/mag/is er?
- afstandsbegrip: wie lijkt op wat?



Het Data mining proces — of beter het **KDD** proces, voor **Knowledge Discovery in Databases** — wordt vaak als volgt opgedeeld:

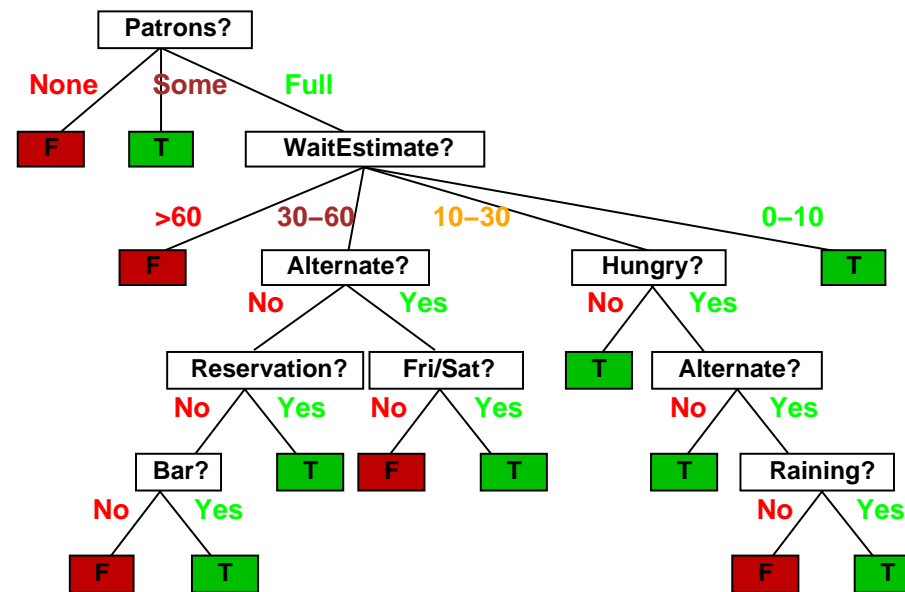
1. data selectie
2. opschonen: de-duplicatie en domein-consistentie
3. verrijking: data-fusie
4. coderen
5. Data mining — het echte gebeuren
6. rapporteren



Er zijn *veel* Data mining technieken, of beter gezegd: algoritmen, die we kunnen gebruiken om data te “minen”:

- (niet vergeten:) statistische methoden
- allerlei “machine learning” technieken uit de AI: evolutionaire algoritmen, neurale netwerken, Bayesiaanse netwerken, ...
- allerlei technieken voor clustering en classificatie: beslissingsbomen, ...
- associatieregels
- ad-hoc methoden

Als voorbeeld iets over **beslissingsbomen** (= **decision trees**). Meestal wordt Quinlan's algoritme J48 gebruikt om uit een aantal voorbeelden zo'n boom te maken. Het berust op Shannon's **informatietheorie**. Interne knopen vragen naar een attribuutwaarde, bladeren geven de classificatie.



Het recursieve algoritme werkt als volgt. Stel we zitten in een knoop, en hebben de verzameling  $V$  van alle voorbeelden die in die knoop terecht komen. Dan:

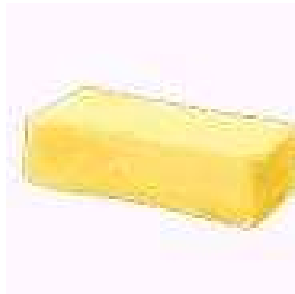
1. Geen voorbeelden meer ( $V = \emptyset$ )? Geef defaultwaarde, bijvoorbeeld “majority-value” van ouderknoop.
2. Alle voorbeelden dezelfde classificatie? Geef die.
3. Geen attributen meer? “Majority-value” van  $V$ .
4. Bepaal “beste” attribuut (**entropie**), splits daarop, en ga recursief verder met de kinderen:

$$\sum_{i=1}^{\nu} \frac{p_i + n_i}{p + n} \cdot \left\{ - \frac{p_i}{p_i + n_i} \log_2 \frac{p_i}{p_i + n_i} - \frac{n_i}{p_i + n_i} \log_2 \frac{n_i}{p_i + n_i} \right\}$$

Ter inspiratie bekijken we nu drie thema's:

- associatieregels
- boodschappenmandjes
- bio-informatica

## Associatieregels



We zijn geïnteresseerd in verbanden tussen verzamelingen “items”: producten, of moleculen, of woorden, of bezoeken aan websites. Regels zijn: “als je boter koopt, koop je meestal ook brood”.

Stel we hebben een database met records = transacties = klanten, waarbij ieder record uit een aantal items = producten bestaat. De **support** van een stel artikelen is het aantal klanten dat al die artikelen koopt, meestal als percentage van het totaal. Een stel artikelen met hoge support (boven een zekere drempel) heet **frequent**.

Bijvoorbeeld: 20% van de klanten in een supermarkt kopen brood, boter en kaas — en wellicht meer producten.

Stel dat van de  $\alpha$  klanten die  $A$  kopen, er  $\beta$  ook  $B$  kopen. We zeggen dan dat de **associatieregel**  $A \Rightarrow B$  een **betrouwbaarheid** heeft van  $\beta/\alpha$ .

Bijvoorbeeld: het zou kunnen zijn dat 80% van de klanten die boter en kaas kopen, ook brood aanschaffen.

We zijn benieuwd naar regels  $A \Rightarrow B$  met zowel hoge betrouwbaarheid als hoge support, dat wil zeggen hoge support voor  $A$  en  $B$  samen (20% in ons voorbeeld).



Er is veel nagedacht over associatieregels:

- geef efficiënte algoritmen om ze te vinden
- hoe vind je vervolgens de interessante?
- en hoe ga je om met niet 0/1-en?

Voor dat laatste gebruikt men fuzzy logica, met in plaats van alleen 0/1 (niet-kopen vs. kopen) ook tussenliggende waarden. Voorbeeld: iemands leeftijd kan 0,3 “jong” zijn.

product = item klant = transactie	1	2	3	4	5	6	7	8	9
1	1	1	0	0	1	1	1	1	0
2	1	0	1	0	0	0	0	1	1
3	0	1	1	0	1	0	1	0	0
4	1	0	1	0	1	1	0	1	1
5	0	0	0	0	1	0	0	0	0
6	0	1	0	0	1	0	1	0	0

Zelfs voor deze kleine database is het moeilijk om te zien dat  $\{2, 5, 7\}$  het enige drietal is dat “gekocht” wordt door minstens 50% (een vaste drempelwaarde) van de klanten.

Frequente itemsets leveren associatieregels:  $\{2, 7\} \Rightarrow \{5\}$ .

De meeste snelle implementaties zijn gebaseerd op:

een deel van iets frequents is zelf ook frequent!

Dit heet de **Apriori-eigenschap**, omdat het wordt gebruikt in **Apriori**, een van de meest bekende Data mining algoritmen.

Kleine frequente verzamelingen worden samengevoegd tot grotere. De databases worden opgeslagen in **FP-bomen**.

## Boodschappenmandjes



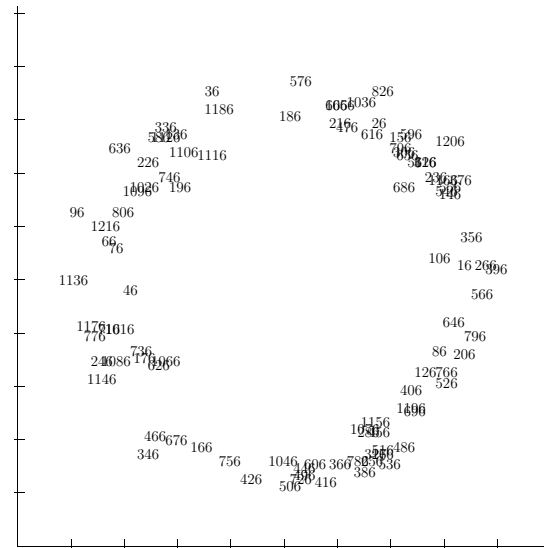
Winkels analyseren klantgedrag: **boodschappenmandjes**.

Toepassingen zijn bijvoorbeeld:

- direct marketing  
“Welke klant doen we een speciale aanbieding?”
- Wat verkopen we?  
“Welke producten zetten we in een kleine winkel, zeg op een station?”
- clustering en classificatie  
“Kunnen we klanten groeperen of herkennen aan hun koopgedrag?” (Ja!)

Een wat algemener doel is het *begrijpen* van klanten.

Een grote winkelketen wilde inzicht in hun “positionering”, gebaseerd op weekverkopten. Dit resulteert in plaatjes als:



Hier zien we 100 winkels.

Om te clusteren heb je een **afstandsbegrip** nodig. Stel dat twee winkels de afgelopen week dit verkocht hebben:

wijn	brood	kaas	tomaat	banaan
7	120	34	0	40
10	100	21	0	0

Hun afstand kan dan zijn:  $3 + 20 + 13 + 0 + 40 = 76$ , of  $3 + 20 + 13 = 36$ , of  $3/10 + 20/120 + 13/34$ , of ...

Het lijkt redelijk te normaliseren voor de totale verkoop bij een winkel. En  $100 \leftrightarrow 120$  verschilt van  $1 \leftrightarrow 21$ .

Hoe dan ook, er zijn veel mogelijkheden!

Clustering technieken kunnen

- hierarchisch zijn: voeg kleinere clusters samen tot grotere
- niet-hierarchisch zijn: verbeter bestaande clustering

Wij gebruiken AI-technieken als Self Organizing Maps (Kohonen's SOMs), gebaseerd op neurale netwerken.

Er zijn **random** aspecten — kansen spelen een rol.





In een supermarkt zijn er *veel* klanten die een *grote* keuze hebben. Associatieregels kunnen eenvoudig worden toegepast.

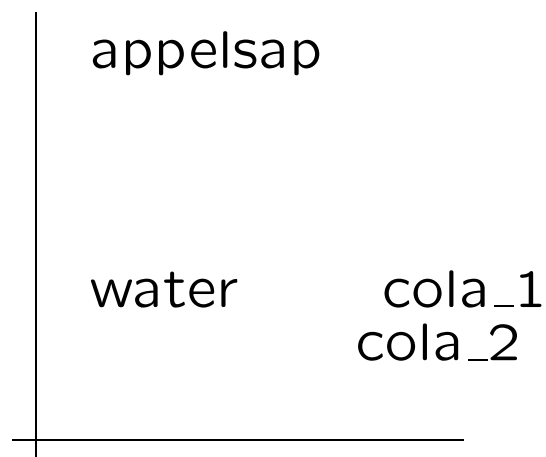
“Onderzoek” onthulde dat vloeier en tabak vaak samen worden verkocht (wat een verrassing!), maar ook dat speciale shag speciale vloeier vereist.

Er is veel interesse in **hierarchieën**: merken ↔ algemene categorieën.

Het luiers—bier verhaal is een sprookje.

Een laatste toepassing in een supermarkt is het volgende. We willen dranken zodanig groeperen dat producten dicht bij elkaar komen als ze elkaar kunnen “vervangen”.

Deze informatie haal je uit de boodschappenmandjes: zulke producten worden zelden samen verkocht. Technieken als boven leiden tot plaatjes als:



## Bio-informatica

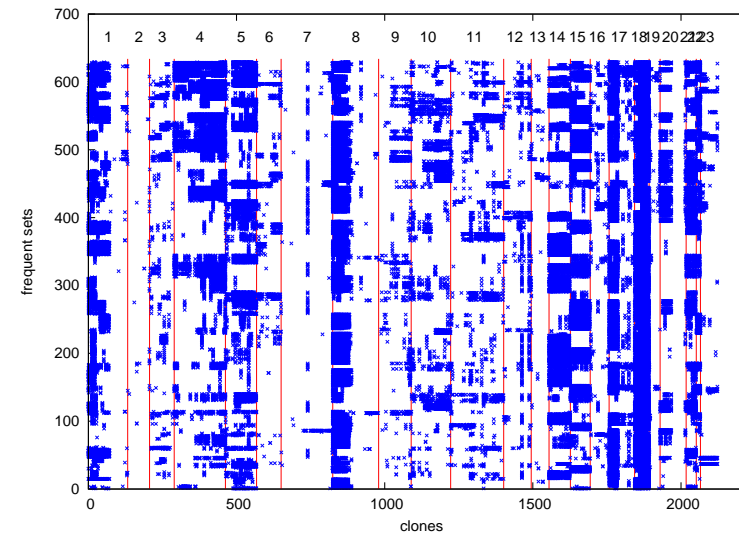
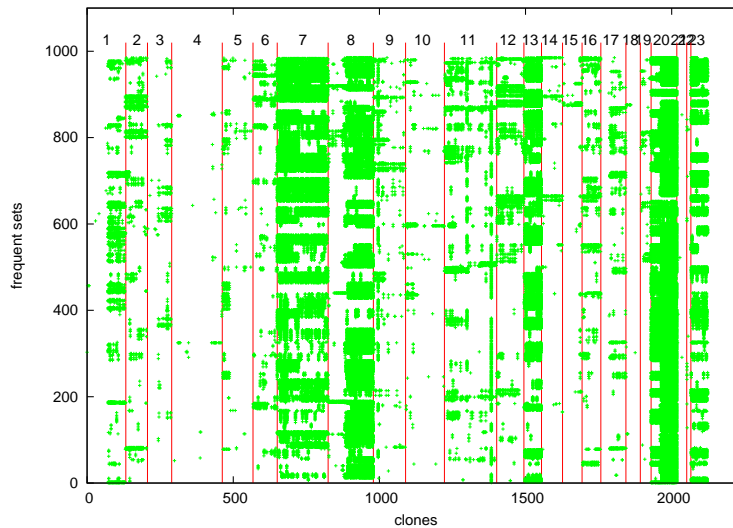


Veel vragen uit de Bio-informatica kunnen als Data mining problemen worden beschouwd. Een enkel voorbeeld.

**DNA** kan gezien worden als een rijtje letters uit het alfabet  $\{A, C, G, T\}$ : *AGGTCAAT...TT*. Menselijk DNA bevat ongeveer 3.000.000.000 letters — het **menselijk genoom**, verdeeld over zo'n  $20^+$  chromosomen.

Stel dat we voor zo'n  $2.000^+$  speciale stukjes DNA (**clones** geheten), netjes verspreid over het menselijk genoom, weten hoeveel er aanwezig is in 150 patiënten. Hier is “hoeveel” een getal tussen  $-5$  (“verlies”) en  $+5$  (“versterking”). Hoe “mine” je deze berg aan data?

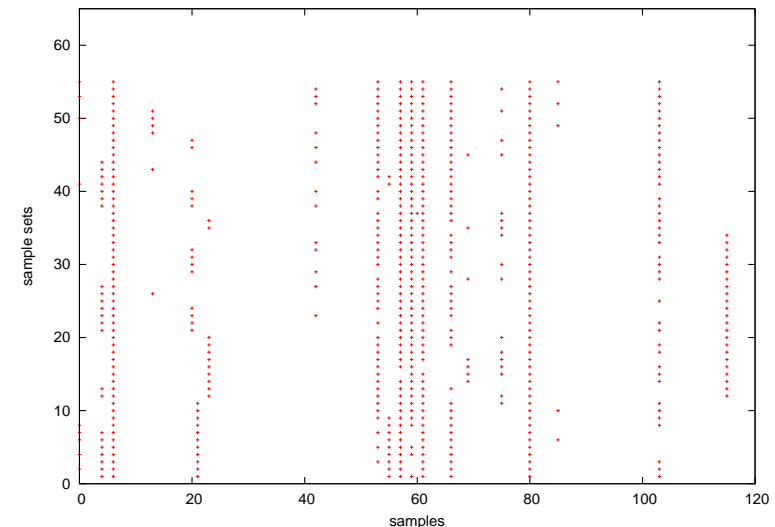
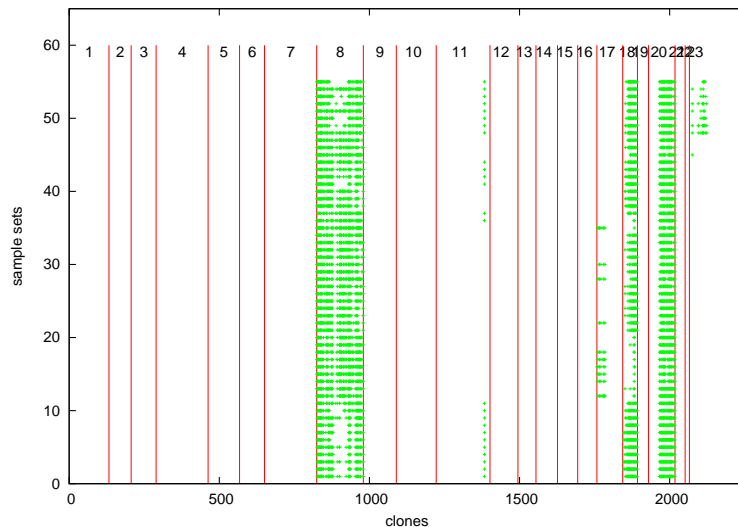
Links: **versterkingen** (985 frequente “duos” ’); rechts: **verliezen** (629 frequente “duos” ’). Verticale **lijnen** zijn de chromosoomgrenzen. Drempelwaarde: 100.



In zo'n plot zien we talrijke groepen patiënten, waarbij elke groep zich ongeveer hetzelfde gedraagt op de 2000<sup>+</sup> clones.

Een tweetal patiënten is frequent als beiden op minstens 100 dezelfde clones een versterking, respectievelijk een verlies hebben.

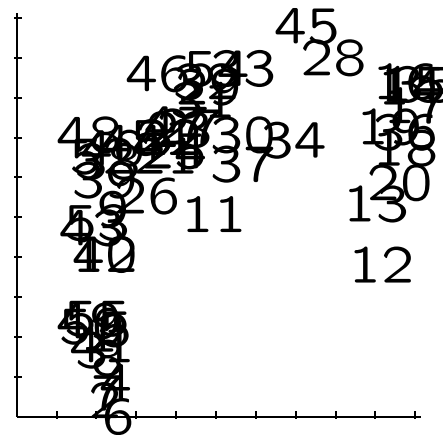
Links: de 55 “maximale” 10-itemsets (met steeds 10 patiënten), gecombineerde versterkingen en verliezen, met drempelwaarde 100; rechts: de patiënten uit die verzamelingen.



De verzamelingen hebben veel gemeen: goed of slecht?



Deze 55 verzamelingen kun je inbedden in het eenheids-vierkant:



De Euclidische afstand lijkt op de afstand die je krijgt door versterkingen en verliezen “kwadratisch op te tellen”.

- beoordeel grenzen versterking/verlies (0.225? fuzzy?)
- maten voor interessantheid, en afstanden ✓
- visualisaties (SOMs, . . . )
- verwerk “aaneengesloten zijn” in support ✓
- voeg gegevens als leeftijd toe
- en tijdsafhankelijkheid; en regelmatigheid ✓
- tool

We zien succesvolle inzet van Data mining.

Aandachtspunten zijn onder meer:

- privacy
- presentatie van resultaten
- toepasbaarheid van mooie algoritmen
- grootschaligheid

Enkele recente afstudeeronderwerpen (**bachelor**, **master**):

- ✘ ✘ Poker; Rummy; Nurikabe; Sudoku; Netflix; Sokoban
- ✘ A Model of Evolution Focussing on Neural Networks
- ✘ Particle Swarm Optimization: Finding Optimal Poker Strategies
- ✘ A Way to Improve the Accuracy of Mining Fused Data
- ✘ Making the Right Offer to the Right Customer
- ✘ Crime Data Mining
- ✘ Unique Factors in the Human Genome
- ✘ Automated Generation and Analysis of Logical Puzzles, The Flats Puzzle
- ✘ Unravelling the Genetic Structure of NP-completeness
- ✘ Computer game development for speech therapy support
- ✘ ✘ Liquid State Machines; Stambomen; Eten; Diagnoses

Voor elke hit op een webserver bevat de accesslog een regel, bijvoorbeeld:

```
68.11.83.42 - - [15/Mar/2006:03:05:46 +0100] "GET /home/kosters/saic/ HTTP/1.1"
200 8307 "http://www.google.com/search?hl=en&lr=&client=firefox-a&rls=org.mozilla
%3Aen-US%3Aofficial&q=sudoku+artificial+intelligence&btnG=Search Mozilla/5.0
(Windows; U; Windows NT 5.1; en-US; rv:1.8.0.1) Gecko/20060111 Firefox/1.5.0.1"
```

Haal van `/scratch/wwwlog` zo'n file, zoals `www.access_log.3.gz` (.gz file onzippen met `gunzip`). Vind nu "iets interessants". Gebruik eventueel Perl ([www.liacs.nl/home/kosters/AI/perl.pdf](http://www.liacs.nl/home/kosters/AI/perl.pdf)).

Deadline: **maandag 14 mei 2007**. Verslag:  $\approx$  3 pagina's, gemaakt met  $\text{\LaTeX}$  ([www.liacs.nl/home/kosters/AI/verslag.tex](http://www.liacs.nl/home/kosters/AI/verslag.tex)); individueel.