

5. KONEOPPIMINEN

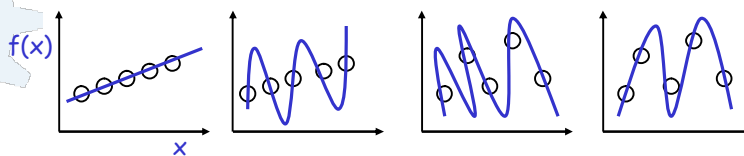
- Älykäs agentti voi joutua oppimaan mm. seuraavia seikkoja:
 - Kuvaus nykytilan ehdoilta suoraan toiminnolle
 - Maailman relevanttien ominaisuuksien päättelyminen havaintojonoista
 - Maailman muuttumisen seuraaminen ja toimintojen vaikutukset
 - Maailman mahdollisten tilojen hyötyarviot
 - Toimintojen arvottaminen
 - Maalitulojen tunnistaminen hyödyn maksimoimiseksi

- Oppijan saama palaute määrää oppimisen tyyppin
 - **Ohjatussa** oppimisessa (supervised l.) tavoitteena on oppia kuvaus syöteiltä tuloksille annettujen esimerkkien perusteella
 - **Ohjaamattomassa** oppimisessa (unsupervised l.) annettuja syötteitä pyritään ryhmittelemään samankaltaisiin ryhmiin ilman erillisiä tulosarvoja
 - **Palauteoppimisessa** (reinforcement l.) oppija saa (ohjattua oppimista epämääräisempää) palautetta, jonka perusteella sen tulee oppia syötteistä
- Tietämyksenesisitys määrää vahvasti oppimisalgoritmin toiminnan

Induktiivinen oppiminen

- Ohjatussa oppimisessä tavoitteena on oppia tuntematon funktio f annettuna *esimerkkejä* $(x, f(x))$, missä x on syöte(vektori)
- *Induktiivisessa inferenssissä* tuloksena on *hypoteesi* h , joka on oppijan approksimaatio f :stä
- Tavoiteltava hypoteesi *yleistää* (generalize) esimerkkien perusteella uusien tapausten ennustamiseksi
- Hypoteesi valitaan hypoteesiluokasta H
- Esimerkiksi kun sekä arvot x ja $f(x)$ ovat reaalilukuja, niin H voi olla mm. korkeintaan astetta k olevat polynomit:

$$3x^2 + 2, x^{17} - 4x^3$$



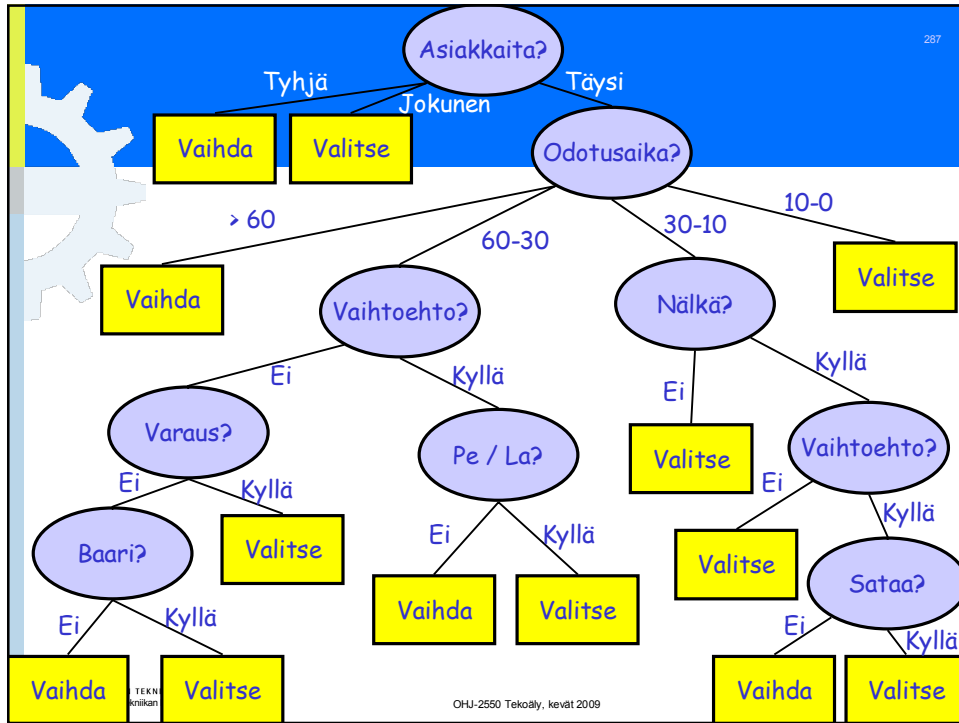
- *Konsistentti hypoteesi* on annetun aineiston kanssa yhtäpitävä
- Mikä valita mahdollisesti monista konsistenteista hypoteeseistä?
- **Occamin partaveitsen** (Occam's / Ockham's razor) mukaan yksinkertaisin hypoteeseistä on paras
- Yksinkertaisuuden määrittäminen?



- Hypoteesiluokka ei aina sisällä konsistenttia hypoteesiä
- Toisaalta (paremman yleistyskyvyn nimissä) voi olla järkevää vaihtaa sovituksen tarkkuutta hypoteesin yksinkertaisuuteen
- Ts., tyytyä huonommin annettuun aineistoon sopivaan hypoteesiin, joka kuitenkin on yksinkertainen
- Hypoteesiluokka on rajoitettava, jotta hyvin aineistoon sopivan hypoteesin löytäminen olisi laskennallisesti tehokasta
- Koneoppiminen keskittyykin melko yksinkertaisten tietämyksen esitysmuotojen oppimiseen

Päätöspuiden oppiminen

- *Päätöspuun* (decision tree) syöte on attribuuttiarvoin ilmaistu olion tai tilanteen kuvaus
- Tuloksena on päätös — ennuste syötettä vastaavalle tulosarvolle
- Jos tulosarvot ovat diskreettejä, niin puu *luokittelee* syötteet
- Jatkuva-arvoisen funktion oppimista nimitetään *regressioksi*
- Syötteen käsittelemiseksi se ohjataan puun juuresta lähtien sisäsolmujen polkua aina lehteen, johon liittyvä päätös on syötteen tulosarvo
- Sisäsolmuihin liittyy attribuuttiarvon testi, jonka mukaan käsiteltävä tapaus ohjataan eteenpäin puussa



288

- (Kohtuullisen kokoinen) päätöspuu on helposti ymmärrettävä tietämyksen esitysmuoto
- Käytännössä tärkeä, heuristisesti opittavissa
- Edellisen esimerkin päätöspuu vastaa päätöstä siitä kannattaako ravintolaan jäädä odottamaan pöytää
- Sen ilmaisema predikaatti on

$$\forall s: \text{Valitse}(s) \Leftrightarrow (P_1(s) \vee \dots \vee P_n(s)),$$
 missä kukin ehtoista $P_i(s)$ on puun polkua juuresta **Valitse**-lehteen vastaavien testien konjunktio
- Eksponentiaalisen kokoisella päätöspuulla voidaan ilmaista mikä tahansa Boolean funktio

TAMPEREEN TEKNILLINEN YLIOPISTO
Ohjelmistotekniikan laitos

OHU-2550 Tekoäly, kevät 2009

16.4.2009

- Yleensä funktion esittämiseen riittää eksponentiaalista kokoa pienempi puu
- Eräät funktiot ovat kuitenkin vaikeita ilmaista päätöspuin, esim. **xor** ja **maj** vaativat eksponentiaalisen kokoisen puun
- Päätöspuut, kuten kaikki muutkin tietämyksen esitysmuodot, soveltuvat hyvin joidenkin funktioiden kuvaamiseen ja huonommin toisten
- n :n muuttujan Booleen funktion totuustaulussa on 2^n riviä, joten eri funktioita on 2^{2^n} kappaletta
- Esim. $n = 6 \Rightarrow 2^{2^6} > 18 \times 10^{18}$, joten konsistentin hypoteesin löytäminen on haastavaa

Päätöspuiden osittava oppiminen

- Algoritmin syöte on *opetusjoukko* (training set), joka koostuu joukosta esimerkkejä (X, y) , missä X on attribuuttiarvojen vektori ja y on arvoihin liitetty luokka-arvo
- Konsistentti puu jossa on oma juuri-lehti-polku kutakin opetusesimerkkiä kohden voitaisiin rakentaa helposti
- Tällöin ei kuitenkaan saavutettaisi yleistyskykyä
- Occamin partaveistä noudattaen meidän tulisi etsiä pienin opetusjoukon kanssa konsistentti puu, mutta pienimmän konsistentin puun löytäminen on laskennallisesti tehotonta (NP-kovaa)

- Menestyksekkäät päätöspuiden oppimisalgoritmit perustuvat heuristiseen pienehkön puun löytämiseen
- Lähtökohtana on valita tärkeimmät attribuutit ensin
- Koska tavoitteena on tapausten luokittelu, niin attribuutti on tärkeä, kun sen vaikutus luokittelussa on suuri
- Päätöspuun muodostaminen itsessään voidaan tehdä rekursiivisella algoritmilla
 - Ensin valitaan puun juureen se attribuutti, joka osoittautuu parhaaksi,
 - jaetaan opetusaineisto tämän attribuutin arvojen perusteella ja
 - jatketaan puun muodostamista alipuista samalla periaatteella

```

Tree growConsTree( Attrs A, Exs S )
{
  if ( all examples in S have class C )
    return an one-leaf tree labeled by C;
  else {
    select an attribute a from A;
    partition S into S1, ..., Sk by the value of a;
    for ( i = 1; i <= k; i++ )
      Ti = growConsTree( A-{a}, Si );
    return a tree T that has a in its root and
      Ti as its i-th subtree; }
}

```



- Jos algoritmi joutuu valitsemaan puun tyhjälle esimerkkien joukolle (opetusaineiston jaon yhteydessä), niin puuksi valitaan lehti, jonka ennustama luokka on koko opetusaineiston yleisin
- Jos taas attribuutit loppuvat kesken s.e. jäljellä olevassa aineistossa on vielä useamman luokan edustajia, niin aineisto on keskenään ristiriitainen i. *kohinainen* (noisy)
- Kohina voi olla seurausta riittämättömistä tilannetta kuvaavista attribuuteista tai se voi olla sovellusalueen aitoa epädeterminismiä
- Yleisimmän luokan ennustaminen on yksinkertainen tapa toimia tässä tilanteessa