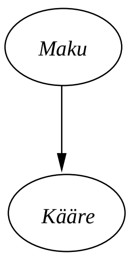
⋆ Menetelmän merkittävä ongelma: jos jotain tapahtumaa ei ole havaittu laisinkaan (pienessä havaintoaineistossa), niin h_{ML} antaa sille nolla-tn.:n

⋆ Muutetaan esim. siten, että riippuen karkin mausta se kääritään probabilistisen säännön perusteella joko punaiseen tai vihreään paperiin



P(salmiakki)
θ

Maku	P(K=p | M)
salmiakki	θ_1
hedelmä	θ_2

⋆ Nyt tn.mallissa on kolme parametria $\theta, \theta_1, \theta_2$

⋆ Bayes-verkon standardisemantiikan perusteella voidaan laskea tapahtumien tn.:iä; esim.

$$\begin{aligned} & P(M = s,\ K = v \mid h_{\theta,\theta_1,\theta_2}) \\ = \ & P(M = s \mid h_{\theta,\theta_1,\theta_2}) \cdot \\ & P(K = v \mid M = s, h_{\theta,\theta_1,\theta_2}) \\ = \ & \theta \cdot (1 - \theta_1) \end{aligned}$$

⋆ Avaamme nyt n makeista, joista s on salmiakkia ja h hedelmää, joiden kääreaperien värien lkm:t ovat p_s, v_s, p_h ja v_h

⋆ Tämän aineiston uskottavuus $P(\vec{d} \mid h_{\theta,\theta_1,\theta_2})$ on

$$\theta^s(1-\theta)^h \theta_1^{p_s}(1-\theta_1)^{v_s}\theta_2^{p_h}(1-\theta_2)^{v_h}$$

⋆ Otetaan logaritmi edellisestä:

$$\begin{aligned} L &= [s \log\theta + h\log(1-\theta)] \\ &+ \ [p_s \log\theta_1 + v_s \log(1-\theta_1)] \\ &+ \ [p_h \log\theta_2 + v_h \log(1-\theta_2)] \end{aligned}$$

⋆ Parametrien suhteen derivoimalla ja hakemalla nollakohta saadaan

$$\frac{\partial L}{\partial \theta} = \frac{s}{\theta} - \frac{h}{1-\theta} = 0 \quad\Rightarrow\quad \theta = \frac{s}{s+h}$$
$$\frac{\partial L}{\partial \theta_1} = \frac{p_s}{\theta_1} - \frac{v_s}{1-\theta_1} = 0 \quad\Rightarrow\quad \theta_1 = \frac{p_s}{p_s+v_s}$$
$$\frac{\partial L}{\partial \theta_2} = \frac{p_h}{\theta_2} - \frac{v_h}{1-\theta_2} = 0 \quad\Rightarrow\quad \theta_2 = \frac{p_h}{p_h+v_h}$$

⋆ θ:n arvo on kuten ennenkin ja θ_1:n arvo on punaisen kääreaperin osuus salmiakkikarkkien joukossa (θ_2 vast.)

⋆ Näin oppimisongelma jakaantuu erillisiksi oppimistehtäviksi kullekin parametrille

Naiivi Bayes

⋆ Luokkamuuttuja C on Bayes-verkon juuri ja attribuutit X_i ovat sen lehtiä

⋆ Naiivi oletus on, että attribuutit ovat ehdollisesti riippumattomia toisistaan annettuna luokka

⋆ Kun käytössä on Boolen muuttujat, niin parametrit ovat

$$\begin{aligned} \theta &= P(C = \mathrm{true}) \\ \theta_{i1} &= P(X_i = \mathrm{true} \mid C = \mathrm{true}) \\ \theta_{i2} &= P(X_i = \mathrm{true} \mid C = \mathrm{false}) \end{aligned}$$

⋆ Näiden arvot löydetään kuten edellä

⋆ Kun verkko on opetettu, niin havainto $\langle x_1, \ldots, x_n \rangle$, jonka luokkamuuttujan C arvo on tuntematon, voidaan luokitella

$$\vec{P}(C \mid x_1, \ldots, x_n) = \alpha \vec{P}(C) \prod_i \vec{P}(x_i \mid C)$$

Jatkuva-arvoiset muuttujat

⋆ Tarkastellaan yhden muuttujan Gaussisen tiheysftion parametrien oppimista

⋆ Aineiston siis tuottaa

$$P(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2} \right)$$

⋆ Mallin parametrit ovat keskiarvo μ ja hajonta σ

⋆ Olk. havaitut arvot $x_1, \ldots, x_n$, nyt uskottavuuden logaritmi on

$$\begin{aligned} L &= \sum_{j=1}^{n} \log \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(x_j-\mu)^2}{2\sigma^2} \right) \\ &= n(-\log\sqrt{2\pi} - \log\sigma) - \sum_{j=1}^{n} \frac{(x_j-\mu)^2}{2\sigma^2} \end{aligned}$$

⋆ Osittaisderivaattojen nollakohdat:

$$\begin{aligned} \frac{\partial L}{\partial \mu} &= -\frac{1}{\sigma}\sum_{j=1}^{n}(x_j - \mu) = 0 \\ &\Rightarrow \ \mu = \frac{\sum_j x_j}{n} \\ \frac{\partial L}{\partial \sigma} &= -\frac{n}{\sigma} + \frac{1}{\sigma^3}\sum_{j=1}^{n}(x_j-\mu) = 0 \\ &\Rightarrow \ \sigma = \sqrt{\frac{\sum_j (x_j-\mu)^2}{n}} \end{aligned}$$

⋆ Siis keskiarvon maksimaalisen uskottavuuden arvo on havaintojen keskiarvo ja hajonnan vastaava arvo on varianssin neliöjuuri

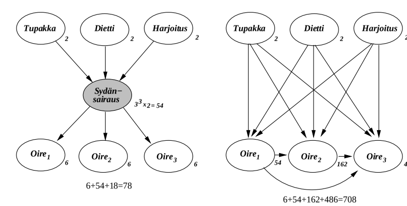
⋆ Jälleen saatiin siis intuitiivisesti oikeat arvot

EM-algoritmi

⋆ Havainnot eivät käytännössä ole täydellisiä, kuten edellä oletettiin

⋆ Sen sijaan on olemassa *piilomuuttujia* l. *latentteja muuttujia*, joiden arvoja ei ole mukana havainnoissa

⋆ Latentin muuttujan ottaminen mukaan Bayes-verkkoon voi vähentää tarvittavien parametrien lkm:ää oleellisesti ja siten helpottaa verkon oppimista



⋆ Kun edellisessä esimerkissä kullakin muuttujalla on 3 mahd. arvoa ovat Bayes-verkkojen parametrien lkm:t 78 ja 708

⋆ Piilomuuttujat kuitenkin hankaloittavat oppimisongelmaa

⋆ Kuinka esim. oppia solmun *Sydänsairaus* ehdolliset tn.:t annettuna sen vanhemmat, koska muuttujan arvoa ei voida havainnoida?

⋆ Sama ongelma koskee oireiden tn.jakaumien oppimista

⋆ EM-algoritmi (expectation-maximization) on yleinen algoritmi tämän ongelman ratkaisemiseksi

Ohjaamaton klusterointi

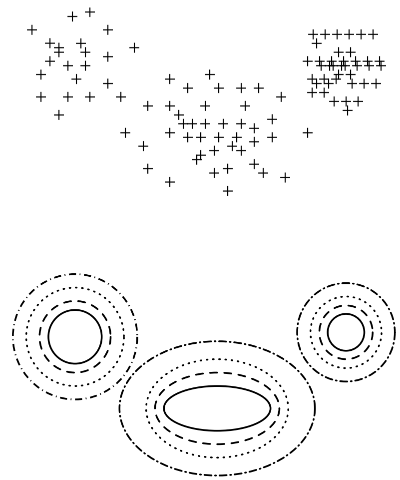
- Tapaukset, joita saamme nähtäväksemme, ovat luokittelemattomia
- Siitä huolimatta usein voidaan tunnistaa tapausluokkia
- Peruslähtökohta luokittelulle on, että havaintoaineiston on tuottanut *sekoitejakauma* P (mixture distr.), jossa on osana k jakaumaa
- Tapauksen generoimiseksi ensin valitaan komponenttijakauma ja sitten arvotaan satunnainen otos valitun jakauman mukaan
- Olk. C valittua komponenttia vastaava satunnaismuuttuja, jonka mahd. arvot ovat $1, \ldots, k$

- Sekoitejakauma on

$$P(\vec{x}) = \sum_{i=1}^{k} P(C = i) P(\vec{x} \mid C = i),$$

missä $\vec{x}$ on tapauksen attribuuttiarvot

- Jatkuva-arvoiselle datalle luonteva valinta komponenttijakaumiksi on monimuuttujainen (multivariate) normaalijakauma
- Sekoitejakauma on siis *mixture of Gaussians*, jonka parametrit ovat
  - komponettijakaumien painot $w_i = P(C = i)$ sekä
  - kunkin komponentin keskiarvo μ_i ja kovarianssi Σ_i
- Tehtävänä on löytää aineiston perusteella sekoitemalli, joka olisi voinut tuottaa havaintoaineiston



- Jos tietäisimme mikä komponenttijakaumista tuotti minkäkin pisteen, voisimme noudattaa normaalia Gaussisen jakauman oppimista
- Jos toisaalta tietäisimme komponenttijakaumien parametriarvot, niin pisteiden (probabilistinen) jakaminen komponentteihin olisi helppoa
- Kumpaakaan ei kuitenkaan tunneta
- EM-algoritmi toimii kuin tuntisimme mallin parametrien arvot ja sen perusteella lasketaan tn.:t kaikille pisteille kaikkiin komponetteihin kuulumiselle
- Sen jälkeen kukin komponentti sovitetaan uudelleen koko aineistoon, pisteet painotettuna ko. komponenttiin kuulumisen tn.:llä

- Edellä mainittuja askeleita toistetaan kunnes menetelmä konvergoituu
- Menetelmä täydentää aineistoa päätellen piilomuuttujien jakaumia nykyisen mallin perusteella
- Piilomuuttajia tässä tapauksessa ovat pisteiden kuulumista komponettijakaumiin indikoivat binääriset satunnaismuuttujat Z_{ij}:

$$Z_{ij} = \begin{cases} 1 & \text{jos } \vec{x}_j\text{:n on tuottanut} \\ & i\text{:s komponettijakauma} \\ 0 & \text{muuten} \end{cases}$$

- Sekoitemallin parametrit alustetaan satunnaisesti ja sen jälkeen toistetaan seuraavia askeleita

1. **E-askel**: Laske todennäköisyydet, että havainnon $\vec{x}_j$ on tuottanut komponenttijakauma i, $p_{ij} = P(C = i \mid \vec{x}_j)$
   - Bayesin kaavan perusteella $p_{ij} = \alpha P(\vec{x}_j \mid C = i) P(C = i)$
   - Tekijä $P(\vec{x}_j \mid C = i)$ on i:nnen Gaussin jakauman tn. pisteessä $\vec{x}_j$
   - $P(C = i)$ on i:nnen jakauman painoparametrin arvo
   - Olk. $p_i = \sum_j p_{ij}$
2. **M-askel**: Päivitä mallin parametrit:

$$\begin{aligned} \mu_i &\leftarrow \sum_j p_{ij} \vec{x}_j / p_i \\ \Sigma_i &\leftarrow \sum_j p_{ij} \vec{x}_j \vec{x}_j^T / p_i \\ w_i &\leftarrow p_i \end{aligned}$$

- E-askel laskee piilomuuttujien Z_{ij} odotusarvot p_{ij}
- M-askel puolestaan hakee mallin parametreille uudet arvot, jotka maksimoivat aineiston log-uskottavuuden annettuna odotusarvot p_{ij}
- EM-algoritmi parantaa aineiston loguskottavuutta jokaisella iteraatiolla
- Joissain tapauksissa EM:n voidaan todistaa saavuttavan uskottavuuden lokaali maksimi (huom. ilman askelkokoa)
- Ongelmatilanteita:
  - Komponenttijakauma supistuu kattamaan vain yhden havainnon, varianssi $= 0 \Rightarrow$ uskottavuus $= \infty$
  - Kaksi komponenttia kattavat samat pisteet (niillä on samat keskiarvot ja varianssit)

- Bayes-verkossa piilomuuttujat ovatkin havaitsemattomien muuttujien arvot kullakin esimerkillä
- Piilo-Markov-mallissa (hidden Markov model, HMM) piilomuuttujia ovat tilasiirtymätn.:t
- EM-algoritmista saadaan eri instantiaatioita erilaisiin malleihin
- Yleisimmässä muodossaan algoritmi typistyy päivityssäännöksi $\vec{\theta}^{(i+1)} =$

$$\arg\max_{\vec{\theta}} \sum_{\vec{z}} P(\vec{Z} = \vec{z} \mid \vec{x}, \vec{\theta}^{(i)}) L(\vec{x}, \vec{Z} = \vec{z} \mid \vec{\theta}),$$

  $\vec{x}$ on kaikki havaitut arvot,
  $\vec{Z}$ on kaikki piilomuuttujat ja
  $\vec{\theta}$ on kaikki parametrit
- Summaus vastaa E-askelta ja maksimointi M-askelta

Tukivektorikoneet

- Annettuna *esimerkerkeistä* koostuva *(opetus)aineisto* S tarkoituksena on oppia *luokittelija* positiivisten ja negatiivisten tapausten erottelemiseksi toisistaan
- Esimerkki muodostuu tapauksesta $\vec{x} \in \mathcal{X}$ ja siihen liittyvästä luokasta $y \in \{-1, 1\}$
- Ennalta kiinnitetty luokittelijoiden joukko $\mathcal{H}$, josta luokittelija h valitaan
- Tarkoitus ei ole luokitella vain S:n alkioita oikein, vaan tavoite on oppia yleisemmin (saman todennäköisyysjakauman) tapausten luokittelu

Lineaariset luokittelijat

- $f(\vec{x})$ on argumenttinsa lineaarinen funktio

$$f(\vec{x}) = \langle \vec{w} \cdot \vec{x} \rangle + b = \sum_{i=1}^{m} w_i x_i + b$$

- Syöte $\vec{x} = (x_1, \ldots, x_m)$ luokitellaan positiiviseksi, jos $f(\vec{x}) \geq 0$ ja muuten $\vec{x}$ luokitellaan negatiiviseksi

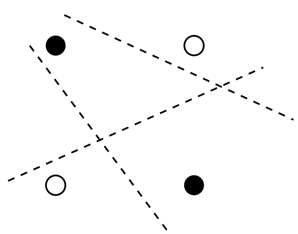
$$\mathrm{sgn}(f(\vec{x})) = \begin{cases} 1, & f(\vec{x}) \geq 0 \\ -1, & \text{muuten} \end{cases}$$

- Yhtälön $\langle \vec{w} \cdot \vec{x} \rangle + b = 0$ määräämä hypertaso jakaa syöteavaruuden kahteen puoliavaruuteen

Lineaariset koneet

- Lineaariset koneet kuten Rosenblatin [1958] Perceptron ovat käsitteellisesti yksinkertaisia
- Lineaarisesti separoituvalle aineistolle Perceptronin voidaan todistaa konvergoivan äärellisellä iteraatioiden lukumäärällä [Novikoff 1962]
- Myöhemmin yksinkertaisia kynnysyksiköitä on keksitty yhdistää neuraaliverkkoon, jolloin muodostuu huomattavasti ilmaisuvoimaisempia luokittelijoita
- Neuraaliverkon opettamiseen ja oppimiseen liittyvät laskennalliset kysymykset ovat vaikeita

Lineaaristen luokittelijoiden rajoitteet: XOR-funktio



Ei ole olemassa lineaarista luokittelijaa, joka erottaisi mustat ja valkoiset pisteet toisistaan

Duaaliesitys

- Kun lähdemme liikkelle nollavektorista, niin lopullinen hypoteesi on opetusesimerkkien lineaarikombinaatio:

$$\vec{w} = \sum_{i=1}^{n} \alpha_i y_i \vec{x}_i$$

- Myös hypoteesin luokittelu voidaan kirjoittaa *duaalikoordinaatistossa*:

$$\begin{aligned} h(\vec{x}) &= \mathrm{sgn}(\langle \vec{w} \cdot \vec{x} \rangle + b) \\ &= \mathrm{sgn}\left(\sum_{i=1}^{n} \alpha_i y_i \langle \vec{x}_i \cdot \vec{x} \rangle + b \right) \end{aligned}$$

Piirrekuvaus

- Opetusaineisto voidaan esittää alkuperäisten muuttujien sijaan niistä johdettuina *piirteinä*

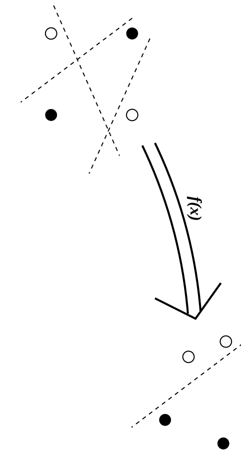
$$\vec{x} = (x_1, \ldots, x_m) \mapsto (\phi_1(\vec{x}), \ldots, \phi_M(\vec{x}))$$

- Piirreavaruus on

$$\mathcal{F} = \left\{ \vec{\phi}(\vec{x}) \mid \vec{x} \in \mathcal{X} \right\}$$

- Useiden — jopa redundanttien — piirteiden käyttäminen helpottaa opittavan käsitteen esittämistä
- Moniulotteinen piirreavaruus ⇒ laskennallinen raskaus

Kuvaus piirreavaruuteen



Kuvataan tapaukset epälineaarisesti (moniulotteiseen) piirreavaruuteen

Ydinfunktiot

- Aineistoon sovelletaan kiinteää epälineaarista muunnosta piirreavaruuteen, jossa lineaarista luokittelijaa käytetään

$$f(\vec{x}) = \sum_{i=1}^{M} w_i \phi_i(\vec{x}) + b$$

- Duaalimuodossa esitetyssä lineaarisessa luokittelijassa lasketaan opetusesimerkkien ja luokiteltavan tapauksen sisätulo

$$f(\vec{x}) = \sum_{i=1}^{n} \alpha_i y_i \langle \phi(\vec{x}_i) \cdot \phi(\vec{x}) \rangle + b$$

- Jos piirreavaruuden sisätulo

$$\langle \phi(\vec{x}_i) \cdot \phi(\vec{x}) \rangle$$

voidaan laskea alkuperäisestä aineistosta, ei eksplisiittistä muunnosta piirreavaruuteen tarvittaisi lainkaan

- *Ydinfunktiolle* K pätee kaikilla $\vec{x}, \vec{z} \in \mathcal{X}$

$$K(\vec{x}, \vec{z}) = \langle \phi(\vec{x}) \cdot \phi(\vec{z}) \rangle ,$$

missä $\phi : \mathcal{X} \to \mathcal{F}$

- Yksinkertaisin ydinfunktio on identtinen kuvaus $K(\vec{x}, \vec{z}) = \langle \vec{x} \cdot \vec{z} \rangle$
- Epälineaarisia kuvauksia puolestaan ovat esim. polynomiset ytimet

$$K(\vec{x}, \vec{z}) = (\langle \vec{x} \cdot \vec{z} \rangle + c)^d$$

- Ydinfunktiota käytettäessä duaalimuodon luokittelu saadaan muotoon

$$f(\vec{x}) = \sum_{i=1}^{n} \alpha_i y_i K(\vec{x}_i, \vec{x}) + b$$

- Tässä esityksessä ei käytetä piirreavaruuden dimensionaalisuutta, joten on mahdollista käyttää suurta piirteiden lukumäärää
- Myöskään epälineaarinen muunnos ϕ ei enää ole eksplisiittisesti näkyvissä

Marginaalit

- Esimerkin $(\vec{x}_i, y_i) \in \mathcal{X} \times \{-1, 1\}$ marginaali funktion $h \in \mathcal{H}$ suhteen on

$$\gamma_i = y_i h(\vec{x}_i)$$

- Jos $\gamma_i > 0$, niin $(\vec{x}_i, y_i)$ on luokiteltu oikein
- Funktion h marginaali aineiston S suhteen on

$$m_S(h) = \min_{(\vec{x}_i, y_i) \in S} \gamma_i$$

- Geometrinen marginaali on normalisoidun luokittelijan marginaali $\gamma = 1 / \|\vec{w}\|_2$

- Olkoon $S = ((\vec{x}_1, y_1), \ldots, (\vec{x}_n, y_n))$ lineaarisesti separoituva aineisto. Hypertaso $(\vec{w}, b)$, joka ratkaisee optimointiongelman

$$\begin{aligned} &\min_{\vec{w}, b} && \langle \vec{w} \cdot \vec{w} \rangle \\ &\text{ehdoilla} && y_i(\langle \vec{w} \cdot \vec{x}_i \rangle + b) \geq 1, \\ & && i = 1, \ldots, n \end{aligned}$$

toteuttaa maksimaalisen marginaalin hypertason geometrisella marginaalilla $1 / \|\vec{w}\|_2$

- Primaalisen optimointiongelman duaaliformulointi antaa seuraavan tuloksen

- Olkoon $S = ((\vec{x}_1, y_1), \ldots, (\vec{x}_n, y_n))$ lineaarisesti separoituva aineisto ja $\vec{\alpha}^*$ kvadraattisen optimointiongelman

$$\begin{aligned} &\text{maksimoi} && W(\vec{\alpha}) = \sum_{i=1}^{n} \alpha_i - \tfrac{1}{2} \sum_{i,j=1}^{n} y_i y_j \alpha_i \alpha_j \langle \vec{x}_i \cdot \vec{x}_j \rangle \\ &\text{ehdoilla} && \textstyle\sum_{i=1}^{n} y_i \alpha_i = 0 \text{ ja } \alpha_i \geq 0,\ i = 1, \ldots, n \end{aligned}$$

ratkaisevat parametrit. Tällöin painovektori

$$\vec{w}^* = \sum_{i=1}^{n} y_i \alpha_i^* \vec{x}_i$$

toteuttaa maksimaalisen marginaalin hypertason geometrisella marginaalilla $1 / \|\vec{w}^*\|_2$

- Ns. Karush-Kuhn-Tucker -lisäehtojen seurauksena vain niillä syötteillä $\vec{x}_i$, jotka ovat lähinnä hypertasoa, voi olla $\alpha_i^* > 0$
- Kaikilla muilla i on oltava $\alpha_i^* = 0$
- Painovektorin lausekkeessa $\vec{w}^* = \sum_{i=1}^{n} y_i \alpha_i^* \vec{x}_i$ riittää ottaa huomioon ne $\vec{x}_i$:t, joilla $\alpha_i^* \neq 0$
- Näitä pisteitä kutsutaan *tukivektoreiksi*, merk. SV:llä kaikkien tukivektoreiden joukkoa
- Optimaalinen hypertaso voidaan ilmaista nyt muodossa

$$f(\vec{z}, \vec{\alpha}^*, b^*) = \sum_{i \in sv} y_i \alpha_i^* \langle \vec{x}_i \cdot \vec{z} \rangle + b^*$$

Olkoon $S = ((\vec{x}_1, y_1), \ldots, (\vec{x}_n, y_n))$ lineaarisesti separoituva ydinfunktion $K(\vec{x}, \vec{z})$ implisiittisesti määräämässä piirreavaruudessa $\mathcal{F}$ ja $\vec{\alpha}^*$ sekä b^* kvadraattisen optimointiongelman

$$\begin{aligned} &\text{maksimoi} && W(\vec{\alpha}) = \sum_{i=1}^{n} \alpha_i - \tfrac{1}{2} \sum_{i,j=1}^{n} y_i y_j \alpha_i \alpha_j K(\vec{x}_i, \vec{x}_j) \\ &\text{ehdoilla} && \textstyle\sum_{i=1}^{n} y_i \alpha_i = 0 \text{ ja } \alpha_i \geq 0,\ i = 1, \ldots, n \end{aligned}$$

ratkaisevat parametrit. Tällöin funktion

$$f(\vec{x}) = \sum_{i=1}^{n} y_i \alpha_i^* K(\vec{x}_i, \vec{x}) + b^*$$

määräämä päätösfunktio $\mathrm{sgn}(f(\vec{x}))$ on ekvivalentti $\mathcal{F}$:n maksimaalisen marginaalin hypertason kanssa. Sen geometrinen marginaali on $\gamma = \left(\sum_{i \in sv} \alpha_i^*\right)^{-1/2}$

- Ydinfunktion on toteutettava ns. Mercer-ehdot, joista seuraa, että optimointiongelma on konveksi
- Näin ollen maksimaalisen marginaalin optimointiongelmalla on yksikäsitteinen ratkaisu, joka voidaan löytää tehokkaasti
- Lokaalien minimien ongelmaa ei esiinny
- Piirreavaruuden määrääminen tekee optimointiongelmasta "helpon"

Yhteenveto

- Tekoäly on todella laaja tutkimuskenttä, myös metodologisesti
- Merkittävää kehitystä tutkimuksessa tapahtuu päivittäin
- Probabilistinen lähestyminen on ajanut ohi loogisen suuntauksen
- Näyttäviä demonstraatioita menetelmien mahdollisuuksista saadaan kiihtyvällä tahdilla
- Myös vähemmälle julkisuudelle jäävät arkisovellukset lisääntyvät
- Fyysisen agentin interaktio toimintaympäristönsä sekä ihmisten kanssa kaipaa vielä edistysaskelia