Viikko 1: Johdantoa

Matti Kääriäinen

matti.kaariainen@cs.helsinki.fi

Exactum C222, 29-31.10.2008.

Tällä viikolla

1. Käytännön järjestelyistä
2. Kurssin sisällöstä ja aikataulusta
3. Johdantoa
   - Mitä koneoppiminen on?
   - Ohjatun oppimisen tehtävistä ja oppimisen mallintamisesta

Käytännön järjestelyistä

- Kurssi koostuu
  - Luennoista
  - Laskuharjoituksista ($1/3$ kokonaispisteistä)
  - Kurssikokeesta ($2/3$ kokonaispisteistä)
- Kurssilla ei ole oppikirjaa, kurssimateriaalina on
  - Luennot
  - Laskuharjoitukset
  - Luentokalvot
  - Kurssin verkkosivulla `http://www.cs.helsinki.fi/group/joko/` mainittu muu materiaali (jos sellaista ilmaantuu)

Viikkoaikataulu

- Luennot salissa C222, ke 12-14 ja pe 12-14
- Laskuharjoitukset salissa C221, ke 10-12 (Mika Urtela)
  - **Ensimmäiset harjoitukset pakolliset**, poikkeuksellisesti mikroluokassa B221
- Viikkoaikataulutavoite:
  - Tehtävät verkkosivulla harjoituksia edeltävän viikon perjantai-iltaan mennessä
  - Viikon luentokalvot verkossa viimeistään samaan aikaan harjoitustehtävien kanssa

Kurssin sisällöstä

1. **Ohjattu oppiminen** (Matti)
   - Johdanto-osuus:
     - Minkälaisista ongelmista ohjatussa oppimisessa ollaan kiinnostuneita
     - Miten oppimistehtävä voidaan määritellä “täsmällisesti”
   - Esimerkkejä koneoppimismenetelmistä joilla oppimistehtäviä voidaan ratkaista
     - k-NN, Naive Bayes, lineaariset menetelmät, päätös- ja regressiopuut
   - Miten valita oikea menetelmä: testaus ja ristiinvalidointi
2. Ohjaamaton oppiminen (Marko)
   - Ryvästystä (eli klusterointia)

Esitiedoista ja -taidoista

- **Matematiikka:** Perustietoja todennäköisyyslaskennasta (ja tilastotieteestä), lineaarialgebrasta, ja analyysistä.
- **Tietojenkäsittelytiede:** Pitää osata kirjoittaa yksinkertaisia ohjelmia ja käyttää laitoksen tietotekniikkaympäristöä. `Matlabia` opetellaan alusta lähtien, ei esitaitovaatimuksia.
- Ensimmäisten viikkojen harjoituksissa palautellaan mieliin matematiikkaa `Matlabin` opettelun yhteydessä

Laskennallinen data-analyysi

- Laaja alue, missä rajat?
  - hahmontunnistus, koneoppiminen, tilastotiede, signaalinkäsittely, merkkijonomenetelmät, tiedon tiivistäminen, . . .
- Tällä kurssilla rajaudutaan käsittelemään koneoppimista, erityisesti ohjatun ja ohjaamattoman oppimisen perusteita.
- Kurssin lähestymistapa on vahvasti tilastollinen, muut lähestymistavat (esim. online-oppiminen, palauteoppiminen, . . .) jäävät muille kursseille

Ohjattu oppiminen

Tavoitteena oppia *ennustamaan* datan perusteella.

- **ennustaa** – esittää (taikuuteen perustuva t. järkiperäinen) arvio tulevaisuudesta, laatia ennuste, “povata”.
  (MOT Kielitoimiston sanakirja 1.0)
- **predict** (engl.) – latinankielisistä sanoista præ=ennen, dicere=sanoa

Tällä kurssilla ennustamisella tarkoitetaan jonkin (kalliin, työlään, hitaan, . . .) vaihtoehtoisen menetelmän tuottaman “oikean tuloksen” ennustamista, ei yleensä tulevaisuuden ennustamista.

Esimerkki 1: merkkien tunnistaminen

- **Syöte:** Merkin kuva (esim. $n \times n$ harmaasävykuva)
- **Ennuste:** Merkin symboliesitys (esim. ASCII-koodi)
- **Tappiofunktio:** $0/1$-tappio: tappio on 0, jos ennuste on sama kuin “totuus”, muuten 1.

Tyypillinen esimerkki *luokitteluongelmasta*: Tavoitteena ennustaa, mihin annetuista luokista syötteenä saatu tapaus kuuluu.

Esimerkki 2: asunnon hinnan ennustaminen

- **Syöte:** asuntoa kuvaavia tietoja (pinta-ala, huoneiden lukumäärä, kunto, osoite, . . .)
- **Ennuste:** asunnon euromääräinen hinta
- **Tappiofunktio:** neliövirhe: tappio on ennusteen ja toteutuneen hinnan erotuksen neliö (?)

Tyypillinen esimerkki *regressio-ongelmasta*: Tavoitteena ennustaa syötteeseen liittyvä (jatkuva-arvoinen) luku

Esimerkki 3: luonnollisen kielen kääntäminen

- **Syöte:** pätkä tekstiä espanjaksi
- **Ennuste:** syötetekstin käännös englanniksi
- **Tappiofunktio:** mitta, joka kuvaa sitä kuinka paljon ennustettu käännös poikkeaa ammattikääntäjän tuottamasta käännöksestä

Esimerkki *rakenteisesta oppimisesta*: Tavoitteena ennustaa syötteeseen liittyvä rakenne (tässä sanajono)

Kohti ennustusongelman formalisointia

Ennustusongelma sisältää seuraavat komponentit:

- **Syötteen tyyppi:** miten ennustusongelman tapaukset kuvataan
- **Ennusteen tyyppi:** minkälaisia ennusteita ennustaja saa tuottaa
- **Tappiofunktio:** funktio, joka annettuna ennuste ja “totuus” kertoo, kuinka paljon ennuste meni pieleen.

Ennustaja on ohjelma, joka tuottaa *syötteen* perusteella *ennusteen*.

Tappiofunktio on ennustusongelman olennainen komponentti: Sen avulla voidaan mitata ennusteen “hyvyyttä”, kun ennustajaa sovelletaan syötteeseen johon liittyvä “oikea tulos” tunnetaan.

Ennustaminen datasta

Mistä ennustaja tulee?

- Käsin ohjelmointi voi olla mahdollista, mutta usein hankalaa ja työlästä.
- Ohjelmointivaivaa voidaan toisinaan vähentää ja/tai ennustustarkkuutta parantaa *oppimalla ennustaja datasta*.

Koneoppimisparadigma (ennustajien oppimisessa): Ennustajan sijaan suunnitellaan *koneoppimismenetelmä* eli ohjelma, joka tuottaa datasta ennustajan.

Koneoppimisprosessi

Idealisoitu versio ennustajan oppimisesta:

- Muotoillaan “tosimaailman ongelma” ennustusongelmaksi (syöte, ennusteen tyyppi, tappiofunktio)
- Kerätään ennustusongelmaan liittyvää dataa, joka ohjatussa oppimisessa koostuu $(\text{syöte}, \text{“oikea ennuste”})$-pareista
- Valitaan/suunnitellaan ennustusongelmaan ja dataan sopiva koneoppimismenetelmä
- Annetaan koneoppimismenetelmän tuottaa datasta ennustaja.
- Testataan ennustajaa (käytännössä)

HUOM: Ylläoleva on vain koneoppimisen “vesiputousmalli” – ts. rankka yksinkertaistus todellisuudesta.

Esimerkki 1: merkkien tunnistaminen

- **Data:**
  - Kerätään edustava joukko dokumentteja, joissa esiintyvät merkit haluttaisiin tunnistaa automaattisesti.
  - Valitaan dokumenteista satunnaisesti otos merkkejä, ja tunnistutetaan ne ihmisillä.
  - Muunnetaan merkkien kuvat ja ihmisten antamat luokitukset (harmaasävykuva vektorina, merkin ASCII-koodi)-formaattiin.

Esimerkki 2: asunnon hinnan ennustaminen

- **Data:**
  - Tietoja (lähiaikoina) toteutuneista asuntokaupoista (esim. kiinteistönvälittäjien tietokannoista)
  - Muunnetaan tiedot
    $(\text{asuntoa kuvaava piirrevektori}, \text{toteutunut kauppahinta})$
    -pareiksi.
  - Sopivia piirteitä esim. asunnon pinta-ala, sijainti, rakennusvuosi, huoneiden lukumäärä, kunto, onko putkiremontti tehty, ...

Esimerkki 3: luonnollisen kielen kääntäminen

- **Data:**
  - Kaksikielinen korpus, joka koostuu espanjankielisistä lauseista ja niiden englanninkielisistä käännöksistä
  - Eristetään datasta piirteitä, joiden avulla se voidaan luontevasti kuvata:
    - Piirteet esim. bifraaseja eli (fraasi, fraasin käännös)-pareja
    - Piirrekuvaus: luetellaan, mitkä bifraasit lauseparissa esiintyy
  - Käytännössä lisäksi käytetään yksikielistä kohdekielen korpusta, josta pyritään oppimaan tuottamaan sujuvaa kohdekieltä

Ohjattu vs. ohjaamaton oppiminen

- *Ohjatussa oppimisessa* opetusdata koostuu (syöte, “oikea ennuste”)-pareista. Ennustajien oppiminen on yleensä ainakin osin ohjattua, tällä kurssilla kokonaan.
  - Tällä kurssilla: lähinnä luokittelua ja regressiota, muut ohjatun oppimisen tehtävät (rakenteinen oppiminen, . . .) jäävät muille kursseille
- *Ohjaamattomassa oppimisessa* data koostuu tyypillisesti pelkistä syötteistä, ja tavoitteena on yleensä jokin muu kuin ennustaminen.
  - Tällä kurssilla: lähinnä ryvästystä

Ohjaamattomaan oppimiseen palataan kurssin lopulla, seuraavassa ohjattua.

Kertausta ja notaatiota

- Toistaiseksi on opittu intuitiivisella tasolla mistä ennustamisessa on kyse
- Seuraavaksi täsmennetään jo opittua, ja esitellään jatkossa käytettävää notaatiota

Syötteet

Merkitään joukkoa josta syötteet tulevat symbolilla $\mathcal{X}$.

Esimerkkejä:

- $\mathcal{X} = \mathbb{R}^d$ (jatkuva tapaus) – $\mathcal{X}$ koostuu piirrevektoreista, joissa on d jatkuva-arvoista komponenttia.
- $\mathcal{X} = \mathcal{D}_1 \times \cdots \times \mathcal{D}_d$, missä $\mathcal{D}_i$:t ovat äärellisiä joukkoja (diskreetti tapaus) – $\mathcal{X}$ koostuu piirrevektoreista, joissa on d diskreettiä komponenttia.
- $\mathcal{X} = \mathcal{X}_1 \times \cdots \times \mathcal{X}_d$, missä kukin $\mathcal{X}_i = \mathbb{R}$ tai äärellinen (jatkuvan ja diskreetin sekoitus)

Tällä kurssilla koneoppimismenetelmät käsittelevät aina piirrevektorimuotoista dataa.

Syrjähyppy: Datan esiprosessointi

- Kuten esimerkeistä 1-3 jo nähtiin, yleensä syötteet eivät ole valmiiksi koneoppimiseen soveltuvassa piirrevektorimuodossa.
- Siksi syötteet pitää esiprosessoida:
  - Valitaan syötteitä kuvaavia piirteitä, jotka on “helppo” laskea (tai muuten mitata)
  - Normalisoidaan piirteet sopivasti oppimisen helpottamiseksi
  - Yritetään valita piirrevektoriin vain oppimisen kannalta olennaisia piirteitä, . . .
- Datan esiprosessointi on monimutkainen taiteenlaji ja käytännössä tärkeä koneoppimisen välivaihe, joka sivuutetaan tällä kurssilla – syötteiden oletetaan siis olevan valmiiksi piirrevektorimuodossa.

Oikeat vastaukset ja ennusteet

- Merkitään symbolilla $\mathcal{Y}$ joukkoa, johon syötteisiin liittyvät oikeat vastaukset kuuluvat.
- Luokittelussa $\mathcal{Y}$ on äärellinen, regressiossa $\mathcal{Y} = \mathbb{R}$.
- Merkitään sallittujen ennusteiden joukkoa symbolilla $\mathcal{Y}'$.
- Yleensä $\mathcal{Y}' = \mathcal{Y}$, mutta ei aina – toisinaan esim. ennustetaan "oikean" y:n sijaan "oikean" y:n todennäköisyysjakaumaa.

Tappiofunktio

- Tappiofunktio on kuvaus $L\colon \mathcal{Y} \times \mathcal{Y}' \to \mathbb{R}^+$.
- Esimerkkejä tappiofunktioista (oletetaan $\mathcal{Y}' = \mathcal{Y}$)
  - Luokittelussa tappiofunktiona on usein 0/1-tappio $L_{0/1}$:

$$L_{0/1}(y, y') = 0, \text{ jos } y = y', \text{ ja } 1 \text{ muuten}$$

  - Regressiossa tappiofunktiona käytetään usein neliövirhettä L_2:

$$L_2(y, y') = (y - y')^2$$

Data ja ennustaja

- Ohjatussa oppimisessa opetusdata on joukko

$$S = \{(x_i, y_i) \mid i = 1, \ldots, n\},$$

  missä $(x_i, y_i) \in \mathcal{X} \times \mathcal{Y}$.

- Ennustaja on kuvaus $f : \mathcal{X} \to \mathcal{Y}'$

Ennustajan syötteellä $x \in \mathcal{X}$ antaman ennusteen $f(x) \in \mathcal{Y}'$ tappio on $L(y, f(x))$. Ennuste on hyvä, jos sen tappio on pieni.

Mutta koska *ennustaja* on hyvä?

Puuttuva linkki

- Toistaiseksi seuraavat kysymykset on sivuutettu:
  - Mistä koneoppimismenetelmälle annettava opetusdata tulee?
  - Mistä koneoppimismenetelmän tuottamalle ennustajalle annettavat syötteet tulevat?
  - Miten nämä liittyvät toisiinsa?

  “Oikeat” vastaukset tietysti riippuvat ratkaistavana olevasta oppimisongelmasta.
- Kysymykset on sivuutettu, koska niihin on vaikea vastata – yleispätevää oikeaa vastausta tuskin on olemassa
- Seuraavassa esitettävä tilastollinen koneoppimisen malli tarjoaa kuitenkin useisiin tilanteisiin tyydyttävästi sopivan vastauksen

Oppimisen malli

- Mallinnetaan opittavaa ilmiötä joukon $\mathcal{X} \times \mathcal{Y}$ todennäköisyysjakaumalla D (jota ei tunneta).
- Koneoppimismenetelmälle annettava opetusdata $S = \{(x_i, y_i) \mid i = 1, \ldots, n\}$ on otos jakaumasta D:
  - $(x_i, y_i) \sim D$, (x_i, y_i) riippumattomia

  Opetusesimerkit ovat siis riippumattomia realisaatioita satunnaismuuttujasta $(X, Y) \sim D$.
- Opitulle ennustajalle annettavat syötteet (ja niihin liittyvät "oikeat vastaukset") tulevat samasta jakaumasta D – nekin ovat riippumattomia realisaatioita satunnaismuutujasta $(X, Y) \sim D$

Näitä mallinnusoletuksia yhdessä kutsutaan *iid-oletukseksi* (independent and identically distributed).

Nappi-intuitio oppimismallille

- Jakauma D on läpinäkymättömässä laatikossa, jossa on nappi. Kun nappia painaa, laatikko sylkee ulos D:n mukaan jakautuneen $(\text{syöte}, \text{“oikea vastaus”})$-parin $(x, y) \in \mathcal{X} \times \mathcal{Y}$.
- Riippumattomuus tarkoittaa, että napin painallukset eivät vaikuta toisiinsa.
- Opetusdata generoidaan painamalla nappia n kertaa. Data annetaan koneoppimismenetelmälle, joka tulostaa ennustajan f.
- Ennustajaa f testattaessa painellaan samaa nappia, mutta ennustajalle näytetään laatikon sylkäisemästä (x, y):stä vain x. Ennustaja antaa ennusteen $f(x)$, ja kärsii tappion $L(y, f(x))$.

Pohdintaa oppimismallista

- Opetusdatan ja testiesimerkkien taustalla olevaa ilmiötä mallinnetaan samalla todennäköisyysjakaumalla D – maailma ei siis muutu oppimisen ja testaamisen välissä/aikana
- Koska opetusdata on otos D:stä, koneoppija saa sen kautta tietoa D:n ominaisuuksista – ja sitä kautta mahdollisuuden ennustaa hyvin testivaiheessa, koska iid-oletuksen nojalla myös testidata on otos D:stä
- Taustatieto opittavasta ilmiöstä tai ilmiöstä tehdyt lisäoletukset voidaan sisällyttää malliin oletuksina jakaumasta D

Jakauman D esittäminen paloissa

Olkoon D joukon $\mathcal{X} \times \mathcal{Y}$ jakauma, ja (X, Y) sen mukaan jakautunut satunnaismuuttuja.

- Marginaalijakaumat:
  - D_X: pelkän X:n jakauma
  - D_Y: pelkän Y:n jakauma
- Ehdolliset jakaumat:
  - $D_{Y|X=x}$: Y:n jakauma, kun $X = x$
  - $D_{X|Y=y}$: X:n jakauma, kun $Y = y$

Yhteisjakauma voidaan ajatella näiden tuloksi/sekoitteeksi:
$D_{X,Y} = D_{X|Y} D_Y$ tai symmetrisesti $D_{X,Y} = D_{Y|X} D_X$

Jakauman D palojen tulkinta

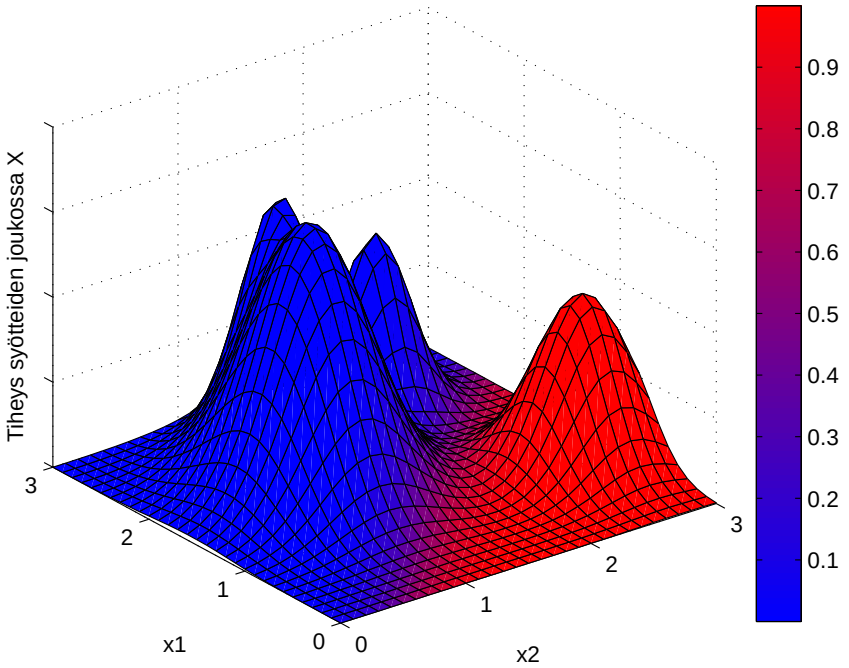
- Marginaalijakauma D_X: pelkkien syötteiden $x \in \mathcal{X}$ jakauma, josta opitun ennustajan syötteet tulevat
- Marginaalijakauma D_Y: pelkkien "oikeiden vastausten" $y \in \mathcal{Y}$ jakauma
- Ehdollinen jakauma $D_{Y|X=x}$: kun tiedetään syötteen olevan x, miten "oikea vastaus" y on jakautunut?
- Ehdollinen jakauma $D_{X|Y=y}$: niiden syötteiden jakauma, joille "oikea vastaus" on y.

Konkreettinen esimerkki jakaumasta D

- Tarkastellaan tilannetta, jossa $\mathcal{X} = [0, 3] \times [0, 3]$, $\mathcal{Y} = \{\text{punainen}, \text{sininen}\}$
- Kyseessä siis luokittelutehtävä
- Seuraavien sivujen kuvat esittävät joukon $\mathcal{X} \times \mathcal{Y}$ todennäköisyysjakaumaa D – ensin yhteisjakaumana, ja sitten “paloiteltuna”

Yhteisjakauma D

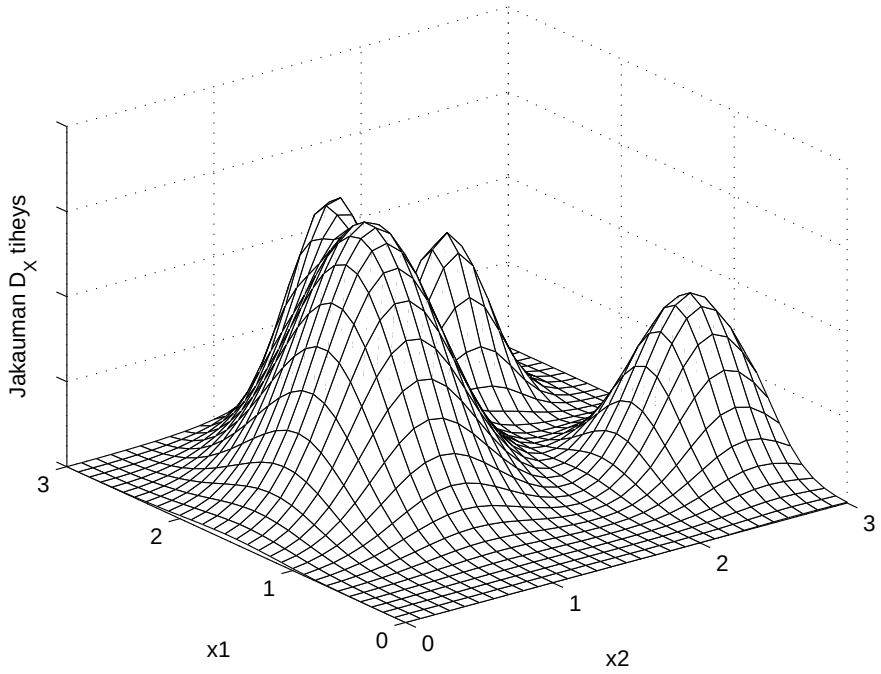
Koko jakauma D.
Kuvaajan korkeus kuvaa sitä, kuinka tyypillinen syöte on. Väri kuvaa todennäköisyyttä kuulua “punaiseen” luokkaan.



Syötteiden jakauma D_X

Syötteiden
jakauma joukossa
$\mathcal{X} = [0, 3] \times [0, 3]$.

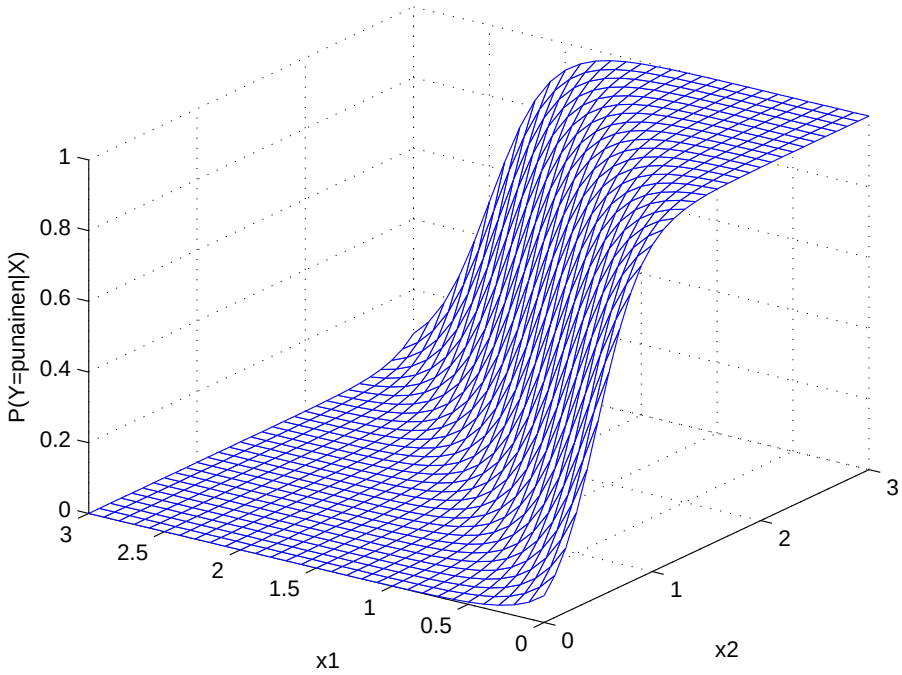
Jakauma
kuvaa sitä minkälaiset
syötteet ovat tyypillisiä
— sekä opetusdatassa
että ennustajan tulevissa
syötteissä.



Luokkien ehdollinen jakauma $D_{Y|X}$

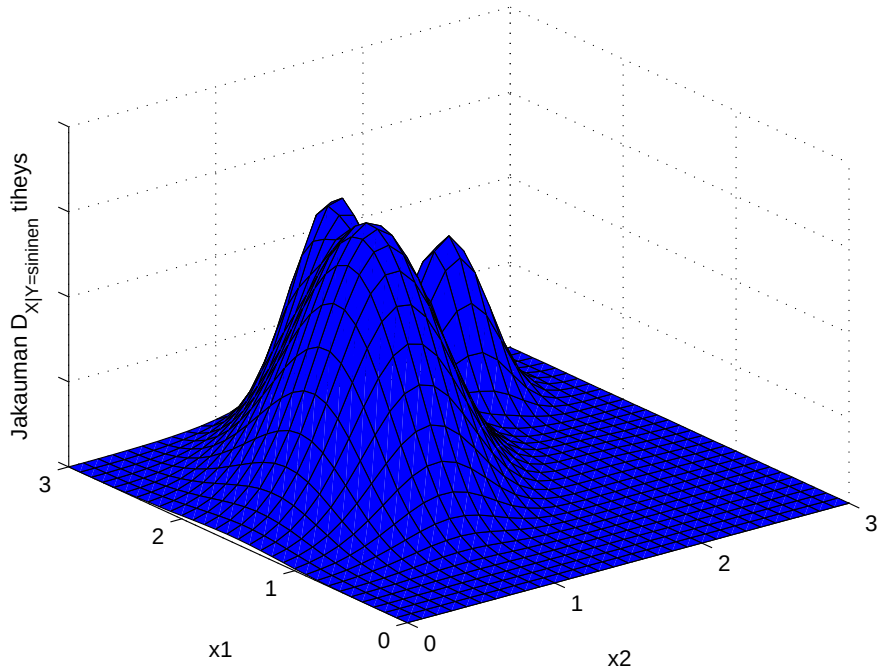
Todennäköisyys, että
luokka on “punainen” kussakin
syöteavaruuden pisteessä
(muuten luokka on “sininen”).

Tässä jakaumassa
ei ole lainkaan tietoa syötteiden
jakaumasta joukossa $\mathcal{X}$.



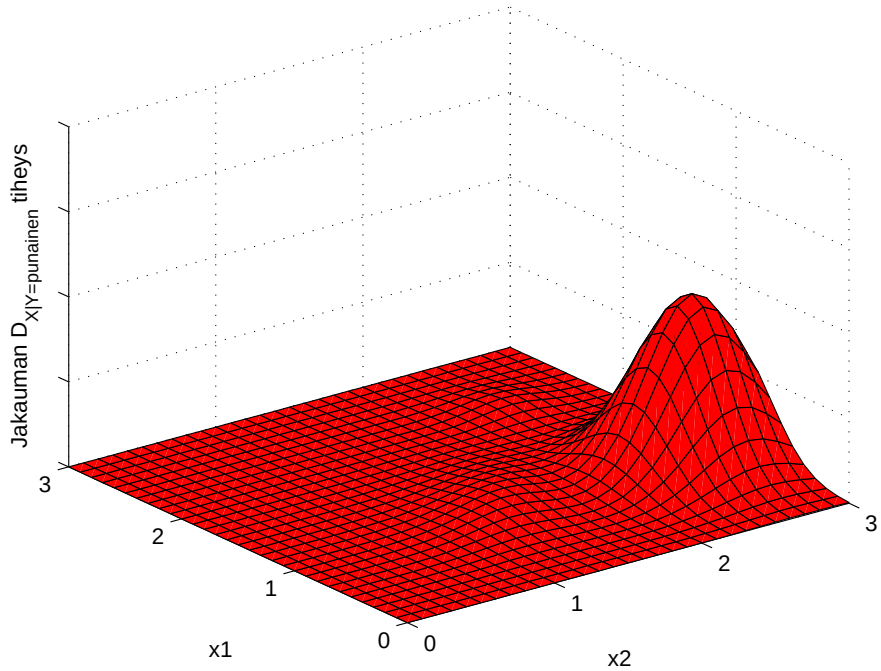
Jakauma $D_{X|Y=\text{sininen}}$

Syötteiden
jakauma luokalle “sininen”.



Jakauma $D_{X|Y=\text{punainen}}$

Syötteiden
jakauma luokalle “punainen”.



Ennustajan oppimistehtävä

Koneoppimismenetelmän tehtävä voidaan nyt määritellä:

- Tulosta opetusdatan avulla ennustaja f, jonka *odotusarvoinen tappio*

$$\mathbb{E}_{(x,y)\sim D}[L(y, f(x))]$$

  on (suurella todennäköisyydellä) pieni.

Idea: Jos iid-oletus pitää paikkaansa, ennustajan f todellisen tappion odotusarvo testiesimerkeillä *on* $\mathbb{E}_{(x,y)\sim D}[L(y, f(x)]$.

Määritelmän mukaan tavoitteena on siis pieni tappio testiesimerkeillä – eli juuri se mitä halutaankin!

Pohdintaa oppimistehtävästä

- Koneoppija voi yrittää oppia opetusdatan avulla approksimaation koko jakaumasta D (generatiivinen oppiminen) tai vain ennustamaan hyvin (diskriminatiivinen oppiminen).
- Koska jakaumaa D ei tunneta, koneoppija ei voi laskea ennustajien odotusarvoista tappiota suoraan. Koneoppiminen ei siis ole vain odotusarvoisen tappion minimointia.
- Pieni odotusarvoinen tappio on jossain mielessä armelias hyvyysmitta koneoppijalle – hyvien ennusteiden antaminen epätodennäköisille $x \in \mathcal{X}$ on vaikeaa, mutta näiden vaikutus otusarvoiseen tappioon on pieni.
- Oppimistehtävässä voi onnistua parhaimmillaankin voin “suurella todennäköisyydellä”, sillä opetusdata on satunnaisotos.

Paras ennustaja*

- Jos jakauma D tunnetaan, paras ennustaja f^* voidaan ainakin periaatteessa ratkaista suoraan kaavalla

$$f^*(x) = \arg\min_{y' \in \mathcal{Y}'} \mathbb{E}_{y \sim D_{Y|X=x}}[L(y, y')].$$

Mutta koska jakaumaa D ei tunneta, koneoppija ei voi suoraan soveltaa yo kaavaa ennustajan löytämäseen.

*Tämän ja kahden seuraavan kalvon kaavoja ei tarvitse muistaa ulkoa, pelkästään ymmärtää.

Paras ennustaja: Luokittelu

- Luokittelutapauksessa parasta ennustajaa f^* sanotaan *Bayes-luokittelijaksi*. Ennustaja f^* määräytyy kaavasta

$$f^*(x) = \arg\max_{y' \in \mathcal{Y}} \mathbb{P}_{y \sim D_{Y \mid X = x}}(y' = y).$$

- Paras ennustaja f^* ennustaa siis kullekin $x \in \mathcal{X}$ sille todennäköisintä luokkaa.

Paras ennustaja: Regressio

- Regressiotapauksessa (tappiofunktiolla L_2) paras ennustaja f^* määräytyy kaavasta

$$f^*(x) = \mathbb{E}_{y \sim D_{Y \mid X=x}}[y].$$

- Paras ennustaja ennustaa siis kullekin $x \in \mathcal{X}$ siihen liittyvän y:n odotusarvoa, so. jakauman $D_{Y \mid X=x}$ odotusarvoa.
- Parasta ennustajaa kutsutaankin y:n ehdolliseksi odotusarvoksi ehdolla x.

Väistämätön tappio

- Parhaankaan ennustajan tappio ei välttämättä ole pieni
- Esimerkiksi luokittelutapauksessa on mahdollista, että $D_{Y|X=x}$ on tasainen jakauma joukossa $\mathcal{Y}$ kaikilla $x \in \mathcal{X}$
- Tällöin parhaankin luokittelijan (Bayes-luokittelijan) odotusarvoinen 0/1-tappio on $1 - 1/|\mathcal{Y}|$
- Yleensä tietysti ollaan kiinnostuneita tapauksista, joissa $D_{Y|X=x}$ keskittyy (useimmiten) vain pieneen $\mathcal{Y}$:n osajoukkoon, ja pienen odotusarvon saavuttaminen on ainakin periaatteessa mahdollista