Viikko 2: Ensimmäiset ennustajat

Matti Kääriäinen

matti.kaariainen@cs.helsinki.fi

Exactum C222, 5.-7.11.2008.

Tällä viikolla

Sisältösuunnitelma:

- Ennustamisstrategioista
- Koneoppimismenetelmiä:
  - k-NN (luokittelu ja regressio)
  - Naive Bayes (luokittelu)

Koneoppimisesta

Kertausta:

- Koneoppijan tehtävänä on tulostaa ennustaja, jonka odotusarvoinen tappio on mahdollisimman pieni
- Parasta ennustajaa ei voi suoraan laskea opetusdatasta, koska jakaumaa D ei tunneta

Miten löytää hyvä ennustaja?

Generatiivinen lähestymistapa

Perusidea:

- Opitaan approksimaatio $\tilde{D}$ jakaumalle D
- Valitaan ennustajaksi

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

  Siis ennustajaksi valitaan paras ennustaja jakaumalle $\tilde{D}$

- Intuitio: Jos jakauma $\tilde{D}$ on “hyvä approksimaatio” oikeasta jakaumasta D, ennustaja f on “hyvä approksimaatio” oikeasti parhaasta ennustajasta f^*.

Ehdollinen lähestymistapa*

- Opitaan approksimaatio $\tilde{D}_{Y|X}$ ehdolliselle jakaumalle $D_{Y|X}$
- Valitaan ennustajaksi

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

  Siis ennustajaksi valitaan paras ennustaja ehdolliselle jakaumalle $\tilde{D}_{Y|X}$ – syötteiden jakauma ei vaikuta parhaaseen ennustajaan.

- Motivaatio: Koska paras ennustaja riippuu vain ehdollisesta jakaumasta $D_{Y|X}$, ei kannata turhaan vaivautua oppimaan ennustamisen kannalta turhaa approksimaatiota koko jakaumalle D.

*Vakiintumaton nimitys, toisinaan tätäkin lähestymistapaa sanotaan diskriminatiiviseksi.

Diskriminatiivinen lähestymistapa

- Diskriminatiivisessa koneoppimisessa yritetään oppia ennustaja suoraan oppimatta ensin approksimoimaan jakaumaa D tai $D_{Y|X}$.
- Motivaatio: Ratkaistaan vain niin monimutkainen ongelma kuin on pakko
- Diskriminatiivisia koneoppimisstrategioita on lukuisia, joista vain joitakin käsitellään tällä kurssilla
- Tärkeän diskriminatiivisen oppimisen strategiaperheen muodostavat seuraavassa esiteltävän *empiirisen riskin minimoinnin* variaatiot

Generatiivinen vs. diskriminatiivinen?

- Generatiivisen (ja ehdollisen) lähestymistavan puolesta puhuu käsitteellinen selkeys ja yksinkertaisuus
- Jakauman D (tai $D_{Y|X}$) oppiminen on kuitenkin periaatteessa vaikeampi tehtävä kuin pelkän hyvän ennustajan löytäminen, mikä puolestaan puhuu diskriminatiivisen lähestymistavan puolesta
- Molemmat lähestymistavat käytännössä tärkeitä, lopulta ratkaistavana oleva ennustustehtävä määrää kumpaa käytetään

Empiirisen riskin minimointi

Ennustajan f *opetusvirhe* (empiirinen riski) on

$$\frac{1}{n}\sum_{i=1}^{n} L(y_i, f(x_i)),$$

missä $\{(x_i, y_i) \mid i = 1, \ldots, n\}$ on opetusdatan joukko.

Opetusvirhe on siis f:n ennusteiden tappion keskiarvo opetusdatalla.

- Empiirisen riskin minimointi: Valitse ennustaja f, jonka opetusvirhe on mahdollisimman pieni.

Idea: Jos opetusvirhe on pieni, voidaan toivoa, että myös odotusarvoinen tappio on pieni – tulevathan opetus- ja testidata samasta jakaumasta. Vaarana kuitenkin ylisovittaminen.

Ylisovittaminen

- Jos f on riippumaton opetusdatasta, pätee

  f:n opetusvirheen odotusarvo = f:n odotusarvoinen tappio.

  Tämä seuraa siitä, että opetus- ja testidata tulevat samasta jakaumasta.

- Jos f riippuu opetusdatasta, ylläoleva ei yleensä pidä paikkaansa
- Erityisesti jos f on valittu empiirisen riskin minimoinnilla, f:n opetusvirhe on yleensä pienempi kuin f:n odotusarvoinen tappio
- *Ylisovittaminen* tarkoittaa tilannetta, jossa opetusdatasta opitun (liian monimutkaisen) ennustajan opetusvirhe on pieni, mutta odotusarvoinen tappio on suuri.

Ensimmäinen koneoppimismenetelmä

Vanha viisaus:

> Entities should not be multiplied unnecessarily.
> William Ockham (1285–1349)

Tässä yhteydessä: Ennen monimutkaisia menetelmiä kannattaa kokeilla, pärjätäänkö yksinkertaisemmilla menetelmillä

- Mikä olisi yksinkertaisin mahdollinen “järkevä” koneoppimismenetelmä?

Taulukointi

Perusidea: ($\mathcal{X}$ ja $\mathcal{Y}$ mielivaltaisia, $\mathcal{Y}' = \mathcal{Y}$)

- Tallennetaan opetusdata taulukkoon
- Tulostetaan ennustaja f, joka syötteellä x toimii seuraavasti:
  - Jos $x = x_i$ jollakin i, ennustetaan y_i. Mikäli $x = x_i$ usealla i, ennustetaan näiden y_i joukossa useimmin esiintyvä y (luokittelu) tai näiden y_i keskiarvo (regressio).
  - Jos $x \neq x_i$ kaikilla i, tulostetaan oletusennuste, esim. koko opetusdatassa useimmin esiintyvä y (luokittelu) tai kaikkien y_i keskiarvo (regressio).

  Näin määräytyvä f on selkeästi kuvaus $\mathcal{X} \to \mathcal{Y}$, siis kelvollinen ennustaja.

Onko taulukointi hyvä lähestymistapa?

Strategia on suoraviivainen toteutus empiirisen riskin minimoinnista: kaikki opetusdata opitaan ulkoa, mutta ei pahemmin yritetä yleistää muille syötteille.

- Taulukointi saattaa toimia hyvinkin, jos opetusesimerkkien joukko kattaa (todennäköisyysmassaltaan) valtaosan syötteiden joukosta $\mathcal{X}$.
- Muuten taulukointi on ongelmissa, sillä ennuste kaikille ennennäkemättömille $x \in \mathcal{X}$ on sama oletusennuste.
- Taulukoinnin tuottaman ennustajan *opetusvirhe* on kuitenkin molemmissa tapauksissa (yleensä) pieni – ylisovittamisen vaara on siis ilmeinen!

Kohti yleistystä

- Taulukointi on menetelmänä sokea syötteiden samankaltaisuudelle: Syötteiden vertailussa käytetään vain mustavalkoista yhtäsuuruutta
- Usein kuitenkin syötteiden yhtäsuuruuden lisäksi voidaan mitata syötteiden välisiä etäisyyksiä
- Seuraavaksi käsitellään lähin naapuri -menetelmiä, jotka voidaan nähdä taulukoinnin etäisyydet huomioon ottavina yleistyksinä

Etäisyysfunktiot

- Etäisyyksiä mitataan *etäisyysfunktiolla** $d\colon \mathcal{X} \times \mathcal{X} \to \mathbb{R}^+$, jolle $d(x, x')$ on sitä pienempi mitä "lähempänä" syöte x' on syötettä x.
- Esimerkkejä etäisyysfunktioista:
  - $\mathcal{X} = \mathbb{R}^d$, $d(x, x') = \sqrt{\sum_{j=1}^{d} (x^j - x'^j)^2}$ (euklidinen etäisyys)
  - $\mathcal{X} = \mathbb{R}^d$, $d(x, x') = \sum_{j=1}^{d} |x^j - x'^j|$ (Manhattan-etäisyys)
  - $\mathcal{X} = \mathcal{X}_1 \times \cdots \times \mathcal{X}_d$, $d(x, x') =$ niiden j lkm, joille $x^j \neq x'^j$ (Hamming etäisyys)
  - $\mathcal{X}$ mielivaltainen, $d(x, x') = 0$, jos $x = x'$, muuten 1 (diskreetti etäisyys)

  Syötteen x komponenttiesitys siis $x = (x^1, \ldots, x^d)$.

*Etäisyysfunktion ei tarvitse olla metriikka, vaikka usein onkin.

Etäisyyspohjaiset koneoppimismenetelmät

Oletus: Etäisyysfunktion d ja tuntemattoman jakauman D välillä on seuraava yhteys:

- Jos $d(x, x')$ on “pieni”, niin $D_{Y|X=x}$ ja $D_{Y|X=x'}$ ovat “lähellä” toisiaan (sopivin adjektiivien “pieni” ja “lähellä” tulkinnoin)

Idea: Jos oletus pitää paikkaansa ja $d(x, x')$ on “pieni”, syötteellä x on järkevää ennustaa suurinpiirtein samoin kuin syötteellä x'.

Etäisyyspohjaisten menetelmien perusidea

- Syötteen $x \in \mathcal{X}$ ennuste lasketaan niiden opetusesimerkkien $(x_i, y_i) \in S$ perusteella, joiden x_i on etäisyysfunktion d suhteen "lähellä" x:ää
- Mitä tarkoittaa "lähellä"? Luonnollisia valintoja ovat mm.:
  - Kiinteä säde $\delta > 0$: x_i on lähellä x:ää, jos $d(x, x_i) \leq \delta$
  - k lähintä naapuria: x_i "lähellä" x:ää, jos sitä lähempänä on alle k muuta opetusesimerkkiä
  - Liukumo: ei jaeta esimerkkejä "lähellä" ja "kaukana" oleviin, annetaan kaikkien x_i:den vaikuttaa x:lle annettavaan ennusteeseen sitä enemmän mitä lähempänä ovat

Seuraavassa tarkastellaan vain lähimmät naapurit -menetelmiä.

k lähintä naapuria -luokittelija

Parametrit: etäisyysfunktio $d\colon \mathcal{X} \times \mathcal{X} \to \mathbb{R}^+$, $k > 0$.

1. Pistä muistiin kaikki opetusesimerkit $(x_1, y_1), \ldots, (x_n, y_n)$
2. Tulosta ennustaja, joka syötteellä $x \in \mathcal{X}$
   (a) Etsii muistista ne opetusesimerkit (x_i, y_i), joille $d(x, x_i)$ on k:n pienimmän joukossa. Merkitään näiden joukkoa symbolilla N_x.
   (b) Tulostaa joukon N_x yleisimmän luokan $y \in \mathcal{Y}$

Miten d valitaan?

- Toiveena, että oletus d:n ja D:n välisestä yhteydestä pätee
- Koska D:tä ei tunneta, oletusta ei tietenkään suoraan voi testata

Usein d:hen yritetään koodata taustatietämystä, tai jos sellaista ei ole, valitaan esim. joku edellä esitetyistä “standardivaihtoehdoista”.

Miten k valitaan?

- Jos k ja siten N_x on liian pieni, ennusteet riippuvat vain muutamasta opetusesimerkistä, ja voivat siten olla epäluotettavia. (ylisovittaminen)
- Jos k on liian suuri, voi olla, että N_x sisältää paljon sellaisia (x_i, y_i), joille $d(x, x_i)$ ei ole "pieni". Tällaiset (x_i, y_i) voivat antaa väärää tietoa jakaumasta $D_{Y|X=x}$. (alisovittaminen)

Sekä d:n että k:n valinnassa kyse *mallinvalinnasta*, josta lisää myöhemmin. . .

k lähintä naapuria -regressio

Parametrit: etäisyysfunktio $d\colon \mathcal{X} \times \mathcal{X} \to \mathbb{R}^+$, $k > 0$.

1. Pistä muistiin kaikki opetusesimerkit $(x_1, y_1), \ldots, (x_n, y_n)$
2. Tulosta ennustaja, joka syötteellä $x \in \mathcal{X}$
   (a) Etsii muistista ne opetusesimerkit (x_i, y_i), joille $d(x, x_i)$ on k:n pienimmän joukossa. Merkitään näiden joukkoa symbolilla N_x.
   (b) Tulostaa

$$f(x) = \frac{\sum_{(x_i, y_i) \in N_x} y_i}{|N_x|}.$$

   Ennuste $f(x)$ on siis x:n lähimpien naapurien keskiarvo.

Painotettu k lähintä naapuria

- Edellä joukon N_x kaikkia alkioita käsiteltiin samalla tavalla muodostettaessa ennustetta $f(x)$
- Intuitiivisesti kuitenkin tuntuisi, että $(x_i, y_i) \in N_x$ on x:ään liittyvän ennusteen kannalta sitä tärkeämpi, mitä pienempi $d(x, x_i)$ on
- Siksi ennustetta muodostettaessa esimerkit usein *painotetaan* etäisyydet huomioivin painoin. Esim. regressiossa voi käyttää vaikkapa painotettua keskiarvoa

$$f(x) = \frac{\sum_{(x_i,y_i)\in N_x} e^{-\alpha d(x,x_i)} y_i}{\sum_{(x_i,y_i)\in N_x} e^{-\alpha d(x,x_i)}}$$

ja luokittelussa vastaavasti painotettua äänestystä. Parametri $\alpha > 0$ määrää, kuinka paljon etäisyys painoon vaikuttaa.

Lähin naapuri -luokittelijoiden (teoreettisia) ominaisuuksia

Kysymys:

- Kuinka hyviä luokittelijoita lähin naapuri -menetelmät tuottavat, jos opetusdatan määrä n on hyvin suuri ($n \to \infty$)?

Seuraavassa oletetaan, että $\mathcal{X} = \mathbb{R}^d$ varustettuna euklidisella etäisyydellä, $\mathcal{Y}$ on kaksialkioinen, ja tappiona on 0/1-tappio.

1 lähin naapuri -luokittelija

Asymptotiikkaa:

- Voidaan osoittaa, että 1 lähin naapuri -ennustajan odotusarvoisen tappion odotusarvo on asymptoottisesti korkeintaan kaksi kertaa niin suuri kuin Bayes-luokittelijan f^* odotusarvoinen tappio
- Siis jos opetusdataa on todella paljon, 1 lähin naapuri -menetelmä on vakiokerrointa 2 vaille paras mahdollinen – kaikilla jakumilla D!

k lähintä naapuria -menetelmän ominaisuuksia

- Voidaan osoittaa, että jos k pidetään kiinteänä opetusdatan joukon koon n kasvaessa, isomman k:n valitsemisesta ei ole asymptoottisesti ainakaan haittaa
- Parhaan asymptoottisen tuloksen saamiseksi k:ta pitää kasvattaa n:n kasvaessa siten, että
  1. $k \to \infty$
  2. $k/n \to 0$

  Tällöin k lähintä naapuria -luokittelijan odotusarvoisen tappion odotusarvo on asymptoottisesti Bayes-luokittelijan odotusarvoinen tappio – jälleen kaikilla jakaumilla D.

Asymptoottisten tulosten tulkinnasta

- Käytännössä opetusdatan koko voi olla suuri, mutta ei koskaan kasva rajatta
- Asymptoottiset tulokset eivät suoraan kerro mitään siitä, miten menetelmät toimivat suurillakaan opetusdatan joukoilla
- Silti asymptoottiset tulokset kertovat menetelmien *konsistenssista* — vaikka asymptoottiset tulokset eivät takaakaan käytännössä mitään, olisi vielä ikävämpää, jos menetelmät eivät toimisi hyvin edes hypoteettisessa “äärettömän suuren” opetusdatan tapauksessa.

k lähintä naapuria -menetelmien aikavaativuudesta

- Suoraviivainen toteutus:
  - Opetusvaiheessa opetusdata tallennetaan taulukkoon.
  - Laskettaessa ennustetta syötteelle x muodostetaan naapurien joukko N_x laskemalla x:n etäisyydet kaikkiin opetusesimerkkeihin
- Jokaiseen ennusteen laskeminen edellyttää siis n etäisyyden laskemisen, eli ennusteen antamiseen kuluva aika kasvaa lineaarisesti opetusdatan koon suhteen.
- Menetelmiä voidaan tehostaa tallentamalla opetusdata hienostuneempiin syöteavaruuden geometriaa hyödyntäviin tietorakenteisiin, joista lähimmät naapurit löytyvät nopeammin.

Naive Bayes

Seuraavaksi jotain ihan muuta:

- Johdatus generatiiviseen lähestymistapaan
- Esimerkkinä Naive Bayes -luokittelija

Kertaus: Generatiivinen lähestymistapa

Palautetaan mieliin, että generatiivisen lähestymistavan perusidea on

1. Oppia opetusdatan perusteella approksimaatio $\tilde{D}$ jakaumalle D
2. Ennustaa kuten $\tilde{D}$ olisi datan generoiva jakauma

Jos opittu $\tilde{D}$ on “hyvä” approksimaatio jakaumasta D, tällä tavalla löydetään “hyvä” ennustaja – ja hyvällä onnella muutenkin. . .

Generatiivinen lähestymistapa

Rajoitutaan tarkastelemaan luokittelua 0/1-tappiolla tapauksessa, jossa syötteiden joukko $\mathcal{X} = \mathcal{X}_1 \times \cdots \times \mathcal{X}_d$ on diskreetti.

Miten jakauman D voisi yrittää oppia opetusdatasta?

- Suora lähestymistapa: Lasketaan opetusdatasta estimaatit $\tilde{P}(X = x, Y = y)$ kaikille todennäköisyyksille $P(X = x, Y = y)$, esim. *suurimman uskottavuuden estimaattorilla*

$$\tilde{P}(X = x, Y = y) = \frac{|\{(x_i, y_i) \in S \mid x_i = x, y_i = y\}|}{|S|}.$$

  Nämä yhdessä määrittävät jakauman $\tilde{D}$.

- Ongelma: periaatteessa mahdollista, mutta estimoitavia todennäköisyyksiä $|\mathcal{X}| \times |\mathcal{Y}| = (|\mathcal{X}_1| \times \cdots \times |\mathcal{X}_d|) \times |\mathcal{Y}|$ kappaletta!

Vaihtoehdoinen strategia jakauman D estimoimiseksi

- Muotoillaan estimointiongelma ennustuksen kannalta edullisempaan muotoon Bayesin kaavan avulla
- Tehdään estimoinnin helpottamiseksi lisäoletuksia (Naive Bayes -oletus)
- Toivotaan, että lisäoletukset ovat niin lähellä totuutta, että niiden varaan rakennettu approksimaatio $\tilde{D}$ johtaa D:n suhteen hyvään ennustajaan

Bayesin kaava

- Palautetaan mieliin *Bayesin kaava*:

$$P(Y = y|X = x) = \frac{P(X = x|Y = y)P(Y = y)}{P(X = x)}$$

  Sanallisesti:

$$\text{posteriori} = \frac{\text{uskottavuus} \times \text{priori}}{\text{normalisointivakio}}$$

- Jakaumia $P(X = x|Y = y)$ on siis yksi jokaiselle $y \in \mathcal{Y}$. Ne kertovat, miten kunkin luokan y syötteet ovat jakautuneet.
- Jakauma $P(Y = y)$ on luokkapriori, joka kertoo, kuinka todennäköinen mikäkin luokka y on ennen syötteen x näkemistä.
- Nimittäjä $P(X = x)$ on kullakin x vakio, joten se voidaan ennustamisen kannalta unohtaa.

Mitä Bayesin kaavalla voitetaan?

- Bayesin kaavan avulla alkuperäinen kysymys “mikä on todennäköisin luokka syötteelle x” saatiin käännettyä ympäri ja paloiteltua:
  - Mikä on todennäköisyys nähdä x, jos luokka on y?
  - Mikä on todennäköisyys nähdä luokka y?

  Toiveena on, että nämä jakaumat $P(X|Y)$ ja $P(Y)$ ovat rakenteeltaan estimoinnin kannalta yksinkertaisempia kuin $P(Y|X)$
- Pyörittely ei kuitenkaan yksin ratkaise estimointiongelmaa: Jakaumat $P(X = x|Y = y)$ ja $P(Y = y)$ ovat tuntemattomia, ja niiden suoraviivainen estimointi edellyttäisi yhteensä $|\mathcal{X}| \times |\mathcal{Y}| + |\mathcal{Y}|$ todennäköisyyden estimoinnin datasta!

Naive Bayes -oletus

- Tehdään (estimointiongelman helpottamiseksi) Naive Bayes -oletuksena tunnettu riippumattomuusoletus:

$$P(X = x|Y = y) = \prod_{j=1}^{d} P(X^j = x^j|Y = y)$$

  Sanallisesti: Syötteen komponentit x^j ovat toisistaan riippumattomia, kun luokka y tunnetaan.

- Koska yleensä on “naiivia” olettaa tuntemattoman jakauman D toteuttavan Naive Bayes -oletuksen, käytännössä yleensä vain pragmaattisesti approksimoidaan D:tä kuten se toteuttaisi oletuksen (vaikka tiedetäänkin ettei oletus pidä paikkaansa)

Mitä lopulta pitää estimoida?

- Yhdistämällä Bayesin kaava ja Naive Bayes -oletus nähdään, että riittää laskea estimaatit seuraaville todennäköisyyksille:
  - $P(X^j = x^j | Y = y)$: $(|\mathcal{X}_1| + \cdots + |\mathcal{X}_d|) \times |\mathcal{Y}|$ estimoitavaa todennäköisyyttä
  - $P(Y = y)$: $|\mathcal{Y}|$ estimoitavaa todennäköisyyttä
- Yhteensä estimoitavia todennäköisyyksiä siis $(|\mathcal{X}_1| + \cdots + |\mathcal{X}_d|) \times |\mathcal{Y}| + |\mathcal{Y}|$
- Tämä on yleensä *paljon* vähemmän kuin $|\mathcal{X}| \times |\mathcal{Y}| = (|\mathcal{X}_1| \times \cdots \times |\mathcal{X}_d|) \times |\mathcal{Y}|$!

Miten jakaumat estimoidaan?

- Jakaumille $P(X^j = x^j | Y = y)$ voidaan käyttää esim. suurimman uskottavuuden estimaattia

$$\tilde{P}(X^j = x^j | Y = y) = \frac{|\{(x_i, y_i) \in S \mid x_i^j = x^j, y_i = y\}|}{|\{(x_i, y_i) \in S \mid y_i = y\}|}$$

- Myös jakaumalle $P(Y = y)$ voidaan käyttää suurimman uskottavuuden estimaattia

$$\tilde{P}(Y = y) = \frac{|\{(x_i, y_i) \in S \mid y_i = y\}|}{|S|}.$$

- Vaihtoehtoisesti voidaan datasta riippumatta olettaa, että (esim.) kaikki luokat ovat yhtä todennäköisiä, ja estimoinnin sijasta käyttää datasta riippumatonta tasaista luokkaprioria $P(Y = y) = 1/|\mathcal{Y}|$.

Estimaattien tasoittaminen

- Suurimman uskottavuuden estimaatit saattavat kärsiä ylisovittamisesta – opetusdatalle annettavat todennäköisyydet ovat suurempia kuin pitäisi
- Tätä voidaan korjata kayttämällä *tasoitettuja estimaatteja*, esim.

$$\tilde{P}(X^j = x^j | Y = y) = \frac{|\{(x_i, y_i) \in S \mid x_i^j = x^j, y_i = y\}| + 1}{|\{(x_i, y_i) \in S \mid y_i = y\}| + |\mathcal{X}_j|}$$

- Seurauksena on erityisesti, että myös ennennäkemättömät tapahtumat saavat nollasta poikkeavan todennäköisyyden

Naive Bayes -luokittelijalla ennustaminen

- Ennustusvaiheessa syötteellä x ennustetaan sitä $y \in \mathcal{Y}$, jolle

$$\tilde{P}(Y = y|X = x) \propto \prod_{i=1}^{d} \tilde{P}(X^i = x^i|Y = y) \times \tilde{P}(Y = y)$$

  on suurin

- Yhtäpitävästi: Ennustetaan sitä $y \in \mathcal{Y}$, jolle

$$\sum_{i=1}^{d} \log \tilde{P}(X^i = x^i|Y = y) + \log \tilde{P}(Y = y)$$

  on suurin. Tämä muoto on käytännössä numeerisen tarkkuuden kannalta parempi.

Naive Bayes: yhteenveto

- Naive Bayes on generatiivinen menetelmä
- Menetelmän ytimessä on yksinkertaistava riippumattomuusoletus: Jakaumaa D approksimoidaan jakaumalla $\tilde{D}$, jossa syötteiden x komponentit x^j ovat toisistaan riippumattomia ehdolla y.
- Oletus on “naiivi” eikä yleensä ole käytännön sovelluksissa perusteltavissa.
- Silti käytäntö on osoittanut, että Naive Bayes toimii usein hyvin – myös silloin, kun riippumattomuusoletus on kaukana totuudesta.