Attribuutin valinta

- Puuhun valittavan attribuutin tulisi parantaa esimerkkien jakoa luokka-attribuutin arvon ennustamiseksi
- Parhaimmillaan attribuutti jakaisi esimerkit joukkoihin, joissa on vain yhden luokan edustajia
- Heikoimmillaan attribuutti ei muuta eri luokkien edustajien suhteellisia osuuksia juuri lainkaan
- Attribuuttien hyödyllisyyden mittaamiseen voidaan käyttää mm. sen antaman *informaation arvoa* eli *Shannon entropiaa*
- Informaatioteoriassa informaatiosisältöä mitataan biteinä
- Yksi bitti riittää kyllä/ei-kysymykseen (kolikonheittoon) vastaamiseksi

- Yleisesti, kun mahdollisilla vastauksilla v_i on todennäköisyydet $P(v_i)$, niin

$$H(P(v_1), \ldots, P(v_n)) = \sum_{i=1,\ldots,n} - P(v_i) \log_2 P(v_i)$$

- Esimerkiksi $H(\frac{1}{2}, \frac{1}{2}) = 2(-\frac{1}{2} \log_2(\frac{1}{2})) = 1$ bitti
- Attribuuttien arvottamiseen sovellettuna haluamme laskea luokka-attribuutin C arvojakaumaan $\underline{P}(C)$ kohdistuvan muutoksen, kun opetusaineisto S jaetaan attribuutin a mukaan osajoukkoihin

$$H_S(\underline{P}(C)) - Gain_S(\underline{P}(C) \mid a),$$

missä

$$Gain_S(\underline{P}(C) \mid a) = \sum_{S_i} (|S_i|/|S|) \cdot H_{S_i}(\underline{P}(C)),$$

kun a jakaa S:n osiin S_i

- Olkoon alkuperäisessä aineistossa S 14 omenaa ja 6 päärynää
- Täten
  $H_S(\underline{P}(C)) = H(0.7, 0.3) \approx 0.7 \times 0.515 + 0.3 \times 1.737 \approx 0.881$
- Jos esimerkiksi attribuutti a jakaa aineiston s.e.

  S_1 = { 7 omenaa, 3 päärynää },
  S_2 = { 7 omenaa } ja
  S_3 = { 3 päärynää }

  niin

  $Gain_S(\underline{P}(C) \mid a) = \sum_{S_i} (|S_i|/|S|) \cdot H_{S_i}(\underline{P}(C))$
  $\approx (10/20) \times H(0.7, 0.3) + 0 + 0$
  $\approx \frac{1}{2} \times 0.881 \approx 0.441$

Oppimisalgoritmin testaaminen

- Jaa esimerkkiaineisto *opetusaineistoksi* ja *testiaineistoksi*
- Sovella oppimisalgoritmia opetusaineistoon, tuota hypoteesi h
- Testaa kuinka suuren osan testiaineiston esimerkeistä h luokittelee oikein
- Em. askelia toistetaan eri opetusaineiston koolla kullakin kerralla vetäen opetusesimerkit satunnaisesti
- Tämän toiminnon kuvaaja on *oppimiskäyrä* (learning curve)
- Vaihtoehtoinen testitapa on *ristiinvalidointi* (cross-validation)

Kohina ja ylisovittaminen

- Jos kaksi esimerkkiä ovat identtiset attribuuttiarvoiltaan, mutta poikkeavat luokka-arvoltaan, niin konsistenttia päätöspuuta ei ole olemassa
- Tällöin ennustetaan lehteen päätyvien esimerkkien yleisintä luokkaa tai palautetaan tieto esimerkkien suhteellisista luokkafrekvensseistä
- Usein kuitenkin voidaan löytää konsistentti päätöspuu vaikka tärkeää tietoa puuttuisikin esimerkeistä
- Oppimisalgoritmi voi käyttää *irrelevantteja* attribuutteja esimerkkien erottamiseen toisistaan, vaikka niiden arvot eivät vaikutakaan tuntemattomaan kohdefunktioon

- Jos esim. nopanheiton silmälukua pyritän ennustamaan sen perusteella
  - minä päivänä ja
  - missä kuussa heitto tapahtui ja
  - minkä värinen noppa on,

  niin kunhan kaksi esimerkkiä eivät ole attribuuttiarvoiltaan identtiset, täsmällinen hypoteesi voidaan löytää
- Löydetty päätöspuu on tietysti aivan väärä hypoteesi
- Mitä useampia attribuutteja on, sitä todennäköisemmin konsistentti hypoteesi löydetään
- Oikea puu olisi yksisolmuinen ennustaja, jossa kunkin silmäluvun edustajien frekvenssi on 1/6
- Tämä on *ylisovittamista* (overfitting), joka vaivaa kaikkia oppimismenetelmiä ja kohdefunktioita, ei pelkästään satunnaisia käsitteitä

Päätöspuun karsinta

- Yksinkertainen tapa vähentää puun ylisovittumista on *karsia* (prune) sitä
- Tarkoituksena on estää jakamasta aineistoa sellaisten attribuuttien perusteella, jotka eivät selkeästi ole relevantteja

- Jos aineisto jaetaan irrelevantin attribuutin perusteella, niin muodostuvissa esimerkkien osajoukoissa on suunnilleen sama luokkajakauma kuin alkuperäisessä aineistossa
- Tällöin informaation lisäys on lähellä nollaa

- Kuinka suurta informaation lisäystä meidän tulisi edellyttää ennen attribuutin hyväksymistä puuhun?

- Tilastollisella *merkitsevyystestillä* testataan kuinka paljon datan antama evidenssi poikkeaa tehdystä *nolla-hypoteesistä*: mitään säännöllisyyttä ei ole havaittavissa
- Jos poikkeama on tilastollisesti epätodennäköinen (yleensä alle 5% todennäköisyys), niin se on vahva evidenssi sen puolesta, että datassa on säännöllisyyttä
- Todennäköisyydet saadaan standardijakaumista, jotka kertovat satunnaisotannalla odotettavissa olevasta poikkeamasta
- Nolla-hypoteesi: attribuutti on irrelevantti ja sen informaation lisäys on nolla (äärettömällä aineistolla)
- On laskettava tn., että nolla-hypoteesin vallitessa v:n esimerkin aineiston poikkeama positiivisten ja negatiivisten esimerkkien odotusarvosta olisi sama kuin opetusaineistosta havaittu

- Olkoot positiivisten ja negatiivisten esimerkkien lukumäärät osajoukoissa s_i ja g_i
- Niiden odotusarvot, kun attribuutti tosiaan on irrelevantti, ovat

$$\hat{s}_i = s \cdot (s_i + g_i)/(s + g)$$
$$\hat{g}_i = g \cdot (s_i + g_i)/(s + g)$$

(s ja g ovat opetusaineiston positiivisten ja negatiivisten esimerkkien kokonaismäärät)

- Poikkeamaa voidaan mitata mm. seuraavasti

$$D = \sum_{i=1,\ldots,v} (s_i - \hat{s}_i)^2/\hat{s}_i + (g_i - \hat{g}_i)^2/\hat{g}_i$$

- Nolla-hypoteesin vallitessa D on jakautunut χ^2-jakauman mukaisesti $(v-1)$:llä vapausasteella
- Tämän jakauman arvot saadaan standarditaulukoista, joten attribuutin irrelevanttius voidaan arvioida

- Edellä esitetty menetelmä on χ^2-esikarsinta
- Karsinta sallii kohinaa opetusesimerkeissä ja pienentää päätöspuita
- Esikarsintaa yleisempiä karsintamenetelmiä ovat jälkikarsinnat, joissa
  - ensin muodostetaan opetusaineiston kanssa niin yhteensopiva puu kuin mahdollista ja
  - sen jälkeen siitä poistetaan ne alipuut, jotka ovat todennäköisesti kohinan aiheuttamia
- Ristiinvalidoinnissa aineisto jaetaan k osaan, joista vuorollaan kutakin käytetään testiaineistona muilla kasvatetulle puulle
- Täten voidaan arvioida puun yleistyskykyä (vs. ylisovittumista)

- Käytännössä päätöspuiden oppimisen on vastattava myös seuraaviin kysymyksiin
  - Puuttuvat attribuuttiarvot: opittaessa ja luokiteltaessa
  - Moniarvoiset diskreetit muuttujat: ryhmittely tai rankaiseminen
  - Numeeriset attribuutit: arvoalueen jako intervalleihin
  - Jatkuva-arvoinen ennustaminen
- Päätöspuut ovat laajasti käytössä ja monia hyviä toteutuksia on tarjolla (ilmaiseksikin)
- Päätöspuut täyttävät ymmärrettävyyden vaatimuksen, joka on amerikkalaiseen lakiinkin kirjattu, toisin kuin esim. neuroverkot

Hypoteesikokoelmien oppiminen

- Valitaankin *kokoelma* (ensemble) hypoteesejä, joiden ennusteet yhdistetään
- Tuotetaan esimerkiksi sata erilaista päätöspuuta saman opetusaineiston perusteella ja annetaan niiden äänestää uuden tapauksen luokasta
- Jos kokoelmassa on 5 hypoteesiä, jotka äänestävät uuden tapauksen luokittelusta, niin väärä luokittelu edellyttää ainakin kolmen hypoteesin virheluokittelua
- Yleisesti ottaen kolmen eri hypoteesin erehtyminen samalla tapauksella on harvinaisempaa kuin yhden ainoan hypoteesin

- Kokoelman hypoteesit eivät ole toisistaan riippumattomia, mutta niiden poikkeamat toisistaan varmistavat, etteivät opetusaineiston virheet voi kopioitua kuhunkin hypoteesiin
- Hypoteesikokoelman käyttö kasvattaa hypoteesien ilmaisuvoimaa ilman laskennallisen vaativuuden räjähtämistä
- Painotetussa opetusaineistossa kuhunkin esimerkkiin liittyy paino $w_j \geq 0$, joka ilmaisee sen merkittävyyttä
- *Oppimisen tehostaminen* (boosting) [Schapire 1990, Freund & Schapire 1996] on yleisin hypoteesikokoelmien oppimismenetelmä
- Opitaan ensin hypoteesi h_1 alkuperäisestä opetusjoukosta, kun kaikkien esimerkkien paino on $w_j = 1$

- h_1 luokittelee osan esimerkeistä oikein ja osan väärin
- Haluaisimme seuraavan hypoteesin luokittelevat paremmin ne esimerkit, joilla h_1 erehtyi
- Oikein luokiteltujen esimerkkien painoa vähennetään ja väärin luokiteltujen painoa kasvatetaan
- Näin muutetun opetusaineiston perusteella tuotetaan hypoteesi h_2
- Hypoteesien oppimista jatketaan samaan tapaan kunnes niitä on ennalta kiinnitetyn parametrin M kertoma lukumäärä
- Lopullinen kokoelmahypoteesi äänestää hypoteesiensä kesken painotetusti niiden opetusjoukolla saavuttaman tarkkuuden suhteessa

Algoritmi AdaBoost(S, A, M)

syöte: S opetusjoukko $(x_1, y_1), \ldots, (x_n, y_n)$, A oppimisalgoritmi, M kokonaisluku, kokoelman koko

```
w ← (1/n, …, 1/n);
for m = 1 to M do
  h[m] ← A(S, w);
  virhe ← 0;
  for j = 1 to n do
        if h[m](xj) ≠ yj then virhe ← virhe + w[j];
  for j = 1 to n do
        if h[m](xj) = yj then w[j] ← w[j]·virhe/(1- virhe);
  w ← Normalisoi(w);
  z[m] ← log (1-virhe)/virhe;          % z[m] on hypoteesin m paino
return PainotettuEnemmistö(h, z);
```


- AdaBoost-algoritmista myönnettiin sen kehittäjille Gödel-palkinto vuonna 2003
- *Heikon oppijan* painotettu virhe opetusaineistolla on vain vähän satunnaista arvausta parempi
- AdaBoost todistettavasti tehostaa heikon oppijan luokittelemaan hypoteesien kokoelmana aineiston virheettömästi (kunhan M on riittävän suuri)
- Usein tehostettu hypoteesien luokka on *yksitasoiset päätöspuut* (decision stumps)
- Hypoteesikokoelman koon nostaminen pudottaa (ainakin alkuun) sekä opetus- että testivirhettä, mutta testivirheen pieneneminen voi jatkua vielä kun opetusvirhe on pudonnut nollaan

5.1 TILASTOLLINEN OPPIMINEN

- Salmiakki- ja hedelmämakeisia on pakattu samanlaisiin käärepapereihin suurissa säkeissä, joissa on seuraavat sekoitussuhteet
  - h_1: 100% salmiakkia
  - h_2: 75% salmiakkia + 25% hedelmää
  - h_3: 50% salmiakkia + 50% hedelmää
  - h_4: 25% salmiakkia + 75% hedelmää
  - h_5: 100% hedelmää
- Satunnaismuuttuja H on säkin tyyppi ja havaintomuuttujia $D_1, \ldots, D_n$ ovat avattujen makeisten maut
- Tehtävänä on seuraavan karkin maun ennustaminen

- Bayes-oppimisessa lasketaan kaikkien hypoteesien todennäköisyys annettuna havaintoaineisto $\mathbf{D}$, jolla on arvo $\mathbf{d}$

$$P(h_i \mid \mathbf{d}) = \alpha\, P(\mathbf{d} \mid h_i)\, P(h_i)$$

- Kun haluamme ennustaa tuntemattoman X arvoa, niin

$$\underline{P}(X \mid \mathbf{d}) = \sum_i \underline{P}(X \mid \mathbf{d}, h_i)\, \underline{P}(h_i \mid \mathbf{d})$$
$$= \sum_i \underline{P}(X \mid h_i)\, P(h_i \mid \mathbf{d})$$

- Edellä jokaisen hypoteesin h_i oletetaan määräävän todennäköisyysjakauman yli X:n
- Ennustukset ovat siis painotettuja keskiarvoja yli yksittäisten hypoteesien ennusteiden

- Bayes-oppimisen keskeisiä arvoja ovat hypoteesien prioritodennäköisyydet $P(h_i)$ ja datan uskottavuus (likelihood) annettuna hypoteesi $P(\mathbf{d} \mid h_i)$
- Olkoot makeissäkkien $h_1, \ldots, h_5$ prioritodennäköisyydet [0.1, 0.2, 0.4, 0.2, 0.1]
- Havaintojen suhteen teemme i.i.d.-oletuksen (independently and identically distributed): kukin havainto on riippumaton muista ja tulee samasta todennäköisyysjakaumasta, joten

$$P(\mathbf{d} \mid h_i) = \prod_j P(d_j \mid h_i)$$

- Jos esim. oikea makeissäkkimme on h_5, jossa on vain hedelmäkarkkeja, niin 10:n ensimmäisen havainnon jälkeen

$$P(\mathbf{d} \mid h_3) = 0.5^{10} \approx 0.001$$

- Koska h_3:lla on korkein prioritodennäköisyys, niin se on alun perin todennäköisin hypoteesi
- Yksi hedelmäkarkin havainnoiminen ei vielä muuta tilannetta, mutta jo kahden peräkkäisen hedelmän jälkeen h_4 muuttuu todennäköisimmäksi hypoteesiksi
- Kolmesta hedelmästä alkaen on (oikea säkki) h_5 kaikkein todennäköisin
- Täten oikea hypoteesi pääsee lopulta dominoimaan ennustusta
- Seuraavan makeisen ennustamisen hedelmäksi todennäköisyys kasvaa monotonisesti kohti arvoa 1 sitä mukaa, mitä useampia hedelmäkarkkeja on havaittu

- Mille tahansa kiinnitetylle priorijakaumalle (joka ei aseta oikean hypoteesin todennäköisyyttä nollaksi) pätee, että väärien hypoteesien posterioritodennäköisyys lopulta katoaa
- Tämä seuraa siitä, että epätyypillisten havaintojen generoiminen jatkuvasti on todennäköisyydeltään katoavan pieni
- Bayes-ennustaminen on optimaalista: mikä tahansa muu menetelmä on oikeassa harvemmin (annettuna samat hypoteesien priorit)
- Hypoteesiavaruudet ovat käytännössä kuitenkin erittäin suuria, jopa äärettömiä
- Summaus (integrointi jatkuvassa tapauksessa) yli hypoteesiluokan ei välttämättä ole laskettavissa

- Usein käytetty approksimointitekniikka on ennustaa sen hypoteesin perusteella, joka on todennäköisin, eli maksimoi arvon $P(h_i \mid \mathbf{d})$
- Tämä on *maximum a posteriori* (MAP) –hypoteesi h_{MAP}
- Kun havaintojen määrä kasvaa, niin MAP-hypoteesin ennuste $P(X \mid h_{MAP})$ ja Bayes-ennuste $P(X \mid \mathbf{d})$ lähenevät toisiaan, koska muiden hypoteesien todennäköisyys putoaa
- Summauksen (tai integroinnin) sijaan nyt ratkaistavaksi jää optimointiongelma
- Esimerkissämme kolmen makeisen jälkeen $h_{MAP} = h_5$ ja neljännen makeisen ennustetaan olevan hedelmäkarkki todennäköisyydellä 1.0, kun oikea Bayes-todennäköisyys olisi 0.8

- Ylisovittumisen estämiseksi Bayes- ja MAP-oppiminen voivat rankaista monimutkaisia hypoteeseja matalalla prioritn.:llä
- Jos esim. H sisältää vain deterministisiä hypoteesejä, niin $P(\mathbf{d} \mid h_i)$ on 1 jos h_i on konsistentti ja 0 muuten
- Tällöin h_{MAP} on Occamin partaveitsen hengessä yksinkertaisin datan kanssa konsistentti looginen teoria

- Toisaalta h_{MAP}:in valitsemiseksi tehtävä arvon $P(\mathbf{d} \mid h_i)\, P(h_i)$ maksimointi on ekvivalenttia sen kanssa, että minimoidaan arvoa $-\log_2 P(\mathbf{d} \mid h_i) - \log_2 P(h_i)$
- Tässä $-\log_2 P(h_i)$ on hypoteesin h_i spesifioimiseksi tarvittavien bittien lukumäärä

- Toisaalta $-\log_2 P(\mathbf{d} \mid h_i)$ on tarvittavien lisäbittien lukumäärä datan määräämiseksi annettuna hypoteesi
- Esim. kun hypoteesi ennustaa datan oikein ($P(\mathbf{d} \mid h_i) = 1$), niin lisäbittejä ei tarvita ($\log_2 1 = 0$)
- Täten MAP-oppiminen valitsee hypoteesin, joka tiivistää datan parhaiten (vrt. Rissasen MDL-periaate)

- Jos hypoteeseille valitaan uniformit priorit, niin MAP-oppiminen typistyy datan uskottavuuden $P(\mathbf{d} \mid h_i)$ maksimoimiseen
- Maksimaalisen uskottavuuden hypoteesin h_{ML} oppiminen on hyvä approksimaatio Bayes- ja MAP-oppimiselle kun dataa on paljon, muttei toimi pienten aineistojen tapauksessa