Luonnollisen kielen tilastollinen käsittely

T-61.281 (3 ov) L

Kevät 2003

Luennot: **Timo Honkela and Kr**
ta Lagus

Laskuharjoitukset: **Vesa Siivola**

Luentokalvot: Krista Lagus ja Timo Honkela

12. Sananmerkitysten yksikäsitteistäminen (word sense disambiguation, WSD)

Ongelman määrittely: Oletetaan sana w, jolle olemassa k erillistä merkitystä $s_1 \ldots s_k$. Tehtävänä on päätellä yksittäisen esiintymän osalta mikä merkitys on kyseessä. Kyse on siis *hahmontunnistuksesta* tai *luokittelusta*.

Hyödyllisiä aineistotyyppejä

- Sense-tagged -korpus: aineisto, johon on jokaisen sanan w esiintymän kohdalle tagattu kyseisen esiintymän merkitys s_i.
  Esim. Senseval (englanninkielinen).
- Sanakirjat ja tesaurukset, ks. esim. sana 'shake' http://www.britannica.c
- Kaksikielinen 'linjattu' aineisto: sama aineisto molemmilla kielillä, johon on merkitty toisiaan vastaavat kohdat, esim. sana tai lause kerrallaan.
- Yksikielinen aineisto, jossa on 'siemeneksi' sense-tagattu (merkitysmerkitty?) pieni osa sanan esiintymistä
- Yksikielinen aineisto, jossa paljon sanan esiintymiä kontekstissa

Korpukset voivat myös olla syntaktisesti tagattuja (sanaluokat jne) tai sisältää pelkän tekstin.

12.1 Eri oppimisperiaatteista

Tunnistusmenetelmät voidaan jakaa ryhmiin mm. oppimisperiaatteen mukaan:

- Ohjaamaton (unsupervised) oppiminen
- Vahvistettu (reinforced) oppiminen
- Ohjattu (supervised) oppiminen

Seuraavaksi tarkastellaan tarkemmin ohjaamatonta ja ohjattua oppimista.

Ohjaamaton oppiminen

- Hahmojen *luokkia ei tiedetä* etukäteen
- Tavoitteena on muodostaa hahmoista ryhmiä, joiden sisällä hahmot ovat samankaltaisia ja joiden välillä on selkeitä eroja (klusterointi)
- Optimoitava funktio on klusteroinnin onnistumista kuvaava mitta
- Aina ei tiedetä edes ryhmien lukumäärää

Ohjattu oppiminen

- Hahmojen *luokat tunnetaan* etukäteen
- Tavoitteena on muodostaa kuvaus piirreavaruudesta luokka-avaruuteen
- Optimoitava funktio perustuu kuvauksessa tapahtuviin virheisiin, ts. pyritään minimoimaan tapahtuvien *luokitteluvirheiden todennäköisyys* tai, mikäli virheisiin liittyy toisistaan poikkeavia kustannuksia, virheiden kokonaiskustannuksen odotusarvo.

Bootstrapping

Luonnollisen kielen aineistoilla relevanttia on lisäksi ns. 'bootstrap' -oppiminen: Pieni osa aineistosta on luokiteltua, jonka avulla päästään alkuun. Tämän jälkeen oppiminen tapahtuu ohjaamattomasti.

12.2 Menetelmien onnistumisen mittaaminen

Keinotekoinen data: pseudosanat

- Menetelmiä voidaan kehittää ja testata keinotekoisella datalla, jonka ominaisuudet varmasti tunnetaan.
- Esim. korvataan kaikki sanojen 'banaani' ja 'ovi' esiintymät pseudosanalla 'banaaniovi'. Mitataan, kuinka hyvin onnistutaan tunnistamaan kutakin 'banaaniovea' vastaava oikea sana.
- Kyseessö on helppo, halpa ja nopea tuottaa laajoja testiaineistoja, joissa tunnetaan sekä disambiguoimaton data että alkuperäinen - oikea - data, joka menetelmän pitäisi löytää.

Onnistumisen laskennalliset ylä- ja alarajat

Mikäli yhteisiä testiaineistoja ei ole, pelkkä numeerinen tulos ei riitä menetelmien onnistumisen mittaamiseen: jotkut ongelmat ovat luonnostaan vaikeampia kuin toiset.

Pyritään siksi hahmottamaan ongelman vaikeus:

- Yläraja 'ground truth': paras mahdollinen tulos. Usein käytetään mittana ihmisen suoriutumista samasta tehtävästä (harvoin 100%).
- Ylärajan määritys tärkeää esim. jos verrataan menetelmien suoriutumista rajallisen mittaisella kontekstilla. Harvoin 100% esimerkiksi, jos ikkuna kovin kapea.
- Alaraja, 'baseline': yksinkertaisin mahdollinen perusmenetelmä. Esim. luokan valinta satunnaisesti tai luokan taustafrekvenssin perusteella.

12.3 Ohjattu disambiguointi

Käytetään notaatiota:

w	monimerkityksinen sana
$s_1 \ldots s_K$	sanan eri merkitykset (senses)
$c_1 \ldots c_I$	sanan w kontekstit korpuksessa
$v_1 \ldots v_J$	piirrejoukko (esim. joukko sanoja), jota käytetään disambiguoinn

Seuraavaksi esitellään joitain ohjatun oppimisen lähestymistapoja, joita on sovellettu merkitysten disambiguointiongelmaan.

Piirteiden valinta

- Yleisesti piirrejoukko vaikuttaa suuresti luokittelun onnistumismahdollisuuksiin.
- Hyvä piirrejoukko on riippuvainen käytetyn luokittimen (tai mallin) ominaisuuksista, eli piirrejoukon valintaa ja mallinnusta ei voida täysin erottaa toisistaan.

Esimerkkejä mahdollisista piirteistä

- tietyn sanan esiintyminen jonkin etäisyyden päässä disambiguoitavasta sanasta, esim. etäisyys(w,'avasi') < 3
- tiettyjen kahden sanan esiintyminen kontekstissa yhdessä
- tietyn sanaluokan tai morfologisen luokan esiintymisfrekvenssi kontekstiikkunassa (jos data on POS- tai morfol. tagattua)
- tietyn sanan tai sanaluokan esiintyminen tietyssä täsmällisessä positiossa suhteessa disambiguoitavaan sanaan (esim. edeltävänä sanana)
- tieto jonkin semanttisen muuttujan, esim. keskustelunaihe, arvosta
- jokin ylläolevien funktio tai yhdistelmä

Bayesläinen luokitin

- Luokitin ei tee piirrevalintaa, vain yhdistää evidenssin eri piirteistä
- Valitaan piirteiksi joukko sanoja
- Luokitin soveltaa *Bayesin päätössääntöä* valitessaan luokan, ts. minimoi luokitteluvirheen todennäköisyyttä:

$$P(s_k|c) = \frac{P(c|s_k)P(s_k)}{P(c)} \tag{1}$$

$P(s_k)$ on merkityksen s_k *prioritodennäköisyys*, eli tn jos emme tiedä kontekstista mitään.

Bayesläinen luokitin: todennäköisimmän luokan valinta

- Jos tehtävänä on vain valita todennäköisin luokka, voidaan jättää kontekstin c todennäköisyys $P(c)$ (joka ei riipu luokasta) laskuissa huomiotta: valitaan merkitys s' jos

$$
\begin{aligned}
s' &= \arg\max_{s_k} P(s_k|c) && (2)\\
&= \arg\max_{s_k} \frac{P(c|s_k)P(s_k)}{P(c)} && (3)\\
&= \arg\max_{s_k} P(c|s_k)P(s_k) && (4)\\
&= \arg\max_{s_k} [\log P(c|s_k) + \log P(s_k)] && (5)
\end{aligned}
$$

Estimointiongelma

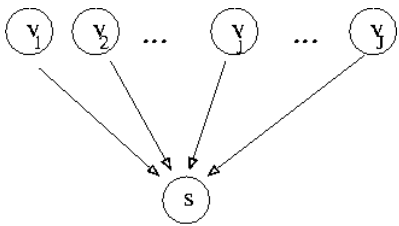
- Käytännön hankaluus: kontekstin piirteiden ehdollisen yhteistnjakauman $P(c|s_k)$ luotettava estimointi tietylle merkitykselle edellyttäisi, että meillä olisi datajoukko, jossa jokainen merkitys esiintyisi kaikissa periaatteessa mahdollisissa konteksteissaan, mieluiten useita kertoja.
- Ratkaisu: helpotetaan estimointia tekemällä sopivia yksinkertaistavia oletuksia (Naive Bayes, Naïve Bayes)

Naive Bayes -luokitin

- *Naive Bayes -oletuksessa* lähdetään siitä, että kukin piirre vaikuttaa luokitukseen toisista piirteistä riippumattomasti:

$$P(c|s_k) = P(v_1, \ldots, v_J|s_k) = \prod_{v_j \text{in} c} P(v_j|s_k) \tag{6}$$

Sama graafisesti:



- Tässä yksinkertaistetussa mallissa (jota kutsutaan myös 'bag of words'-malliksi) sanojen järjestyksellä kontekstissa ei ole merkitystä, ja sama sana voi esiintyä kontekstissa useita kertoja.

Naive Bayes -luokitin, jatkoa

- Sovellettaessa edelliseen päätössääntöön (kaava **??**) saadaan *Naive Bayes päätössääntö*:

$$\text{Valitaan } s' \text{ jos } s' = \arg\max_{s_k}[\log P(s_k) + \sum_{v_j \text{in} c} P(v_j|s_k)] \qquad (7)$$

- Näistä $P(v_j|s_k)$:lle ja $P(s_k)$:lle lasketaan ML-estimaatit luokitellusta opetusdatajoukosta:

$$\begin{aligned} P(v_j|s_k) &= \frac{C(v_j, s_k)}{C(s_k)} \\ P(s_k) &= \frac{C(s_k)}{C(w)} \end{aligned}$$

  jossa C(...) tarkoittaa lukumäärää opetusdatajoukossa

Naive Bayes -luokitin, jatkoa

- Kuuden substantiivin *duty, drug, land, language, position, sentence* luokituksessa kun aineistona oli Hansard-korpus saatiin 90% tunnistustulos (Church, Gale & Yarowsky, 1992).

- Esimerkkejä 'drug'-sanan merkityksille sovelletuista piirteistä:

Merkitys	Piirteet ko. merkitykselle
medication	prices, prescription, patent, increase, consumer, pharmaceutical
illegal substance	abuse, paraphernalia, illicit, alcohol, cocaine, traffickers

Naive Bayes -luokitin, yhteenveto

- Naive Bayes-luokitin on yksinkertainen ja melko robusti; antaa kohtuullisia tuloksia monenlaisissa ongelmissa.
- Naive Bayes -ongelma: epärealistiset riippumattomuusoletukset
- ML-estimoinnin ongelma: käyttää kaikki piirteet, ts. ei kykene tekemään piirrevalintaa

Eräs informaatioteoreettinen lähestymistapa

- Edellisessä mallissa käytettiin kaikki kontekstin sanat estimoinnissa.
- Nyt päinvastainen lähestymistapa: valitaan yksittäinen mahdollisimman hyvä indikaattori, jonka arvo voidaan selvittää kustakin kontekstista kullekin merkitykselle.
- Esimerkki: Ranskan 'prendre'-sanan merkitykset 'tehdä päätös' (*prendre une decision* ja 'ottaa mitta' (*prendre une mesure*, luotettava indikaattori olisi verbin objektina oleva sana.
- Disambiguoitava sana: $w = prendre$
  Valitaan piirrejoukoksi (indikaattoriksi) esim. objektipositiossa olevat sanat $V = \{$*mesure, note, exemple, décision, parole*$\}$
  Käännösten joukko: $K = \{take, make, rise, speak\}$

Menetelmän kuvaus

Maksimoidaan yhteisinformaatio piirteen ja merkityksen välillä. Muistellaan yhteisinformaation kaavaa:

$$I(X;Y) = \sum_{x \in X} \sum_{y \in Y} p(x,y) \log \frac{p(x,y)}{p(x)p(y)}$$

partitiointi = jonkin joukon jako osajoukkoihin.

Flip-Flop -algoritmi

- $K=$ käännösten joukko, $V=$ piirteiden joukko
- Valitse V:lle satunnaisesti jokin partitiointi Y
- Toista, kunnes parannus ei ole enää suuri:
  1. Etsi K:lle partitiointi X siten, että $I(X;Y)$ maksimoituu
  2. Etsi V:lle partitiointi Y siten, että $I(X;Y)$ maksimoituu

Esim. jos partitioidaan vain kahteen ryhmään:
Käännettävä sana: prendre
Mahdolliset käännökset $K = \{take, rise, make, speak\}$
Objektiposition piirteet $V = \{$*mesure, note, exemple, décision, parole*$\}$

Esimerkki

Data:

- 'prendre une mesure ': take measure,
- 'prendre notes': take notes,
- 'prendre exemple': take an example
- 'prendre une decision': 'make a decision'
- 'prendre la parole': 'rise to speak'

Paras partitiointi:

- $X_1 = \{take\}$, $X_2 = \{rise, make, speak\}$
- $Y_1 = \{mesure, note, exemple\}, Y_2 = \{decision, parole\}$

Kommentteja flip-flop -algoritmia koskien

- Jos tehdään täyshaku eli kokeillaan kaikki partitioinnit kummallekin joukolle, K ja V, menee exponentiaalinen aika
- Flip-Flop-algoritmi kuitenkin vie lineaarisen ajan
- Toistetaan Flip-Flop-algoritmi eri indikaattorelle (objekti, määre, edellinen sana, jne); valitaan indikaattori, joka maksimoi yhteisinformaation

Kommentteja flip-flop -algoritmia koskien, jatkoa

Huom. Käännöksistä saadut 'labelit' eivät välttämättä sanan eri merkityksiä vaan osaksi saman merkityksen eri ilmenemismuotoja (ks. sananmuotojen variaatio, esim. 'cent' viitatessaan merkitykseen senttimetri voisi ilmetä suomeksi muodoissa ('cm','sentti'). Partitioidessaan myös eri labelien joukon S algoritmi itse asiassa ryhmittelee nämä eri muodot pienemmäksi joukoksi merkityksiä.

Kääntämisen kannalta katsottuna tehtävä on siis ohjatun oppimisen tehtävä. Merkitysten etsimisen (esim. niiden oikea lukumäärä) ja disambiguoinnin kannalta taas ohjaamatonta oppimista.

12.4 Sanakirjapohjainen disambiguointi

Sanakirjamerkitysmäärittelyihin perustuva menetelmä

Notaatio ja piirteet:

sanan kuvaus	symboli	eri merkitykset	niiden määritelmät
tulkittava sana	w	$s_1 \dots s_K$	$D_1 \dots D_K$
w:n kontekstin sana j	v_j	$s_{j_1} \dots s_{j_L}$	$D_{j_1} \dots D_{j_L}$

Piirrejoukko: $E_{v_j} = \bigcup_{j_i} D_{j_i}$ eli ei välitetä kontekstin sanojen monimerkityksisyyksistä (yhdistetään määritelmät sanan w piirteitä laskettaessa).

Valintakriteeri:

$$s' = \arg\max_{s_k} \bigcap (D_k, E_{v_j})$$

Huom: Tämä on matemaattisesti sama kuin vektoriavaruusmenetelmä binääriarvoisilla, normalisoimattomilla vektoreilla. Myös paremmat samankaltaisuusmitat mahdollisia (piirteiden painotus ja vektorien pituuksien normalisointi).

Sanojen semanttisiin aiheluokkiin perustuva menetelmä

(kirjassa nimellä 'thesaurus-based disambiguation')

- Pohjana yleinen semanttinen luokitus (thesauruksessa, mm. Roget, tai muuten, mm. Longman)
- Luokat t aihealueita (topics),
  esim. {'urheilu', 'sota', 'musiikki', 'kalastaminen',...}
- score= monellako kontekstin sanalla löytyy yhteinen luokka sananmerkityksen s_k kanssa, ts. tässäkään ei käytetä normalisointeja tai painotuksia tai todennäköisyyksiä
- $$\text{score}(s_k) = \sum_{v_j \text{ in } c} \delta(t(s_k), v_j)$$
- $\delta(t(s_k), v_j) = 1$ joss $t(s_k)$ on jokin v_j:n luokista

Sanojen semanttisiin aiheluokkiin perustuva menetelmä, jatkoa

- Olet. että kukin merkitys s_k kuuluu täsmälleen yhteen luokista
- Sanan luokkien joukko on sen eri merkitysten luokkien unioni
  ts. ei yritä disambiguoida kontekstisanojen merkityksiä
- Jättää hyödyntämättä sanat, joita ei etukäteen sem. luokiteltu (esim. uudet sanat, jonain aikakautena kuuluisat henkilönnimet 'Navratilova', 'Jeltsin' jne., jotka voisivat olla oikein hyviäkin piirteitä)

Yarowskyn adaptiivinen versio edellisestä

- Lähtee liikkeelle annetusta semanttisesta luokittelusta ja parantaa sitä luokittelemalla myös uudet sanat niiden kontekstien luokkatiedon perusteella (bootstrapping)
- Yarowskyn kokeissa konteksti = 100 sanan ikkuna
- Soveltaa Naive Bayes -oletusta eli oletetaan piirteiden vaikutus riippumattomiksi:

$$\begin{aligned} P(t_l|c_i) &= \frac{P(c_i|t_l)}{P(c_i)}P(t_l) \\ &= \frac{\prod_{v \text{ in } c_i} P(v|t_l)}{\prod_{v \text{ in } c_i} P(v)}P(t_l) \end{aligned}$$

- Käytetään suurehkoa kynnysarvoa α hyväksymään luokka vain, jos se on riittävän todennäköinen tälle kontekstille (eli jos $P(t_l|c_i) > \alpha$, suuri α).

Yarowskyn algoritmi

1. **Luokittele kontekstit** sanojen luokkien perusteella
   päivitä $P(t_l|c_i)$ kaikille c_i
   päivitä $t(c_i) = \{c_i : n \text{ luokat, joille } P > \alpha\}$

2. **Luokittele sanat** kontekstien luokkien perusteella
   $V_j = \{\text{kontekstit, joissa piirre } v_j\}$
   $T_l = \{\text{kontekstit, joilla luokkana } t_l\}$
   päivitä $P(v_j|t_l) = \frac{|V_j \bigcap T_l|}{\sum_j |V_j \bigcap T_l|}$
   päivitä $P(t_l) = \frac{|V_j \bigcap T_l|}{\sum_l \sum_j |V_j \bigcap T_l|}$

3. **Disambiguoi** moniselitteinen sana w

$$
\begin{aligned}
s' &= \arg\max_{s_k} \prod_{v_j \text{ in } c} P(t(s_k), v_j) \\
&= \arg\max_{s_k} \prod_{v_j \text{ in } c} P(t(s_k)) P(v_j | t(s_k)) \\
&= \arg\max_{s_k} [\log P(t(s_k)) + \sum_{v_j \text{ in } c} \log P(v_j | t(s_k))]
\end{aligned}
$$

Yarowskyn algoritmi, jatkoa

- uusille sanoille: estimoidaan luokat
- vanhoille sanoille: päivitetään luokat
- salliiko algoritmi vanhan sanan luokan poistumisen? (kirjassa ei kerrota...)
- muuten toimitaan todennäköisyksillä, mutta kontekstien ja piirteiden luokitus on 'kova'.
- menetelmällä on saatu hyviä tuloksia testidatalla
- topic-lähestymistapa toimii hyvin silloin, kun merkitys aihealuekohtainen. Kuitenkaan kaikilla sanoilla merkitykset eivät riipu aihealueista → huonoja tuloksia.

2-kielisen aineiston käännöksiä hyödyntävä menetelmä

- tarvitaan: 2-kielinen sanakirja + toisen kielen korpus
- käännetään sana kaikilla eri tavoilla (kaikilla eri merkityksillä)
- käännetään sanan kontekstipiirre (yksinään, olet. vain yksi käännös)
- tarkastellaan käännösparien keskinäisiä frekvenssejä toisen kielen korpuksessa. Mikäli jokin vaihtoehto on riittävän todennäköinen, valitaan sen sisältämä tulkinta (testataan merkitsevyys, ja valitaan vain mikäli $p >$ esim. 90%).
- yleistyy suoraviivaisesti monen piirteen tutkimiselle

Yksi merkitys per aihe, yksi merkitys per kollokaatio

- Sanakirjapohjaiset menetelmät tarkastelivat jokaista sananesiintymää erillisenä
- Kuitenkin esiintymien välillä keskinäisiä riippuvuuksia
- Oletus 1: Yksi merkitys per aihe: sanalla yleensä yksi merkitys läpi koko dokumentin
- Oletus 2: Yksi merkitys per kollokaatio: sanan merkitys riippuu vahvasti aivan lähikontekstin sanoista, mukaanlukien sanojen järjestys ja sanaluokkatieto (ts. usein toisistaan täysin erilliset merkitykset ovat eri syntaktisessa ja/tai semanttisessa roolissa ympäristön suhteen)

Yarowskyn algoritmin ongelmia

NB:n tekemä piirteiden riippumattomuusoletus. Vaihdetaan siksi mallia: valitaan 1 'paras' piirre ja käytetään vain sen mukanaantuoma evidenssi. (Kolmas vaihtoehto olisi mallittaa oikeasti piirteiden yhteisvaikutus)

Yarowskyn (toinen) algoritmi

1. Sovella oletusta 1: sovella sanakirjamenetelmää tai bootstrappaystä merkitysten valintaan, mutta nyt vain 'parasta' diskriminoivaa kontekstin piirrettä käyttäen
2. Sovella oletusta 2: 'äänestetään' sanan merkitys dokumentin sisällä kaikille samaksi

Yarowsky: 2-vaihe parantaa tuloksia 27%. Lopulliset tulokset luokkaa 90%-96%.

12.5 Ohjaamaton merkitysten ryhmittely

- Edelläkuvatut menetelmät tarvitsevat leksikaalisia resursseja: sanakirjoja tai (pieniä) merkityksin tagattuja aineistoja jokaiselle monimerkityksisen sanan eri merkitykselle
- Aina sellaisia ei ole, esim. erikoistermien tai uusien merkitysten ilmaantuessa.
- Jos disambiguoinnilla tarkoitetaan merkitysten taggausta, täysin ohjaamattomasti ei voida disambiguoida.
- Voidaan kuitenkin klusteroida sanan esiintymät, ja toivoa/olettaa että kukin klusteri vastaa sanan yhtä merkitystä.
- Hyvä tai huono puoli: ryhmittely voi olla tarkempaa kuin esim. sanakirjoissa.
- Menetelmiä esim. EM-algoritmi, k-means, SOM, hierarkkiset klusterointimenetelmät, ...

EM-algoritmi disambiguoinnissa

Seuraavassa esitellään EM (Expectation Maximation) -algoritmin käyttö disambiguonnissa.

Esitetyissä kaavoissa K on eri merkitysten lukumäärä. $c_1, \ldots, c_i, \ldots, c_I$ ovat monitulkintaisen sanan kontekseja korpuksessa. $v_1, \ldots, v_j, \ldots, v_J$ ovat sanoja, joita käytetään piirteinä disambiguoinnissa.

EM-algoritmi disambiguoinnissa, alustusvaihe

- Alusta mallin μ parametrit satunnaisesti.
  Parametrit ovat
  - $P(v_j|s_k), 1 \leq j \leq J, 1 \leq k \leq K$ ja
  - $P(s_k), 1 \leq k \leq K$.

  Laske log-likelihoodin arvo korpuksesta C annettuna malli μ. Arvo saadaan kertomalla keskenään yksittäisten kontekstien c_i todennäköisyydet $P(c_i)$, missä $P(c_i) = \sum_{k=1}^{K} P(c_i|s_k)P(s_k)$:

  $$l(C|\mu) = log \prod_{i=1}^{I} \sum_{k=1}^{K} P(c_i|s_k)P(s_k) = \sum_{i=1}^{I} \sum_{k=1}^{K} P(c_i|s_k)P(s_k)$$

EM-algoritmi disambiguoinnissa, EM-osuus

Niin kauan kuin $I(C|\mu$:n arvo kasvaa, toistetaan E- ja M-askeleita.

- E-askel:

  Kun $1 \leq j \leq J, 1 \leq k \leq K$, estimoidaan h_{ik} eli posterioritodennäköisyys sille, että s_k generoi c_i:n seuraavasti:

  $h_{ik} = \frac{P(c_i|s_k)}{\sum_{k=1}^{K} P(c_i|s_k)}$

  Jotta $P(c_i|s_k)$ saadaan lasketuksi, tehdään Naive Bayes -oletus:

  $P(c_i|s_k) = \prod_{v_j \in c_j} P(v_j|s_k)$

EM-algoritmi disambiguoinnissa, M-askel

- M-askel:

  Estimoidaan uudelleen parametrit $P(v_j|s_k)$ ja $P(s_k)$ käyttämällä ML-estimointia:

  $$P(v_j|s_k) = \frac{\sum_{i=1}^{I} \sum_{c_i:v_j \in c_i} h_{ik}}{Z_j}$$

  Kaavassa $\sum_{c_i:v_j \in c_i}$ laskee summan kaikkien sellaisten kontekstien yli, joissa kontekstin sana v_j esiintyy. $Z_j = \sum_{k=1}^{K} \sum_{i=1}^{I} \sum_{c_i:v_j \in c_i} h_{ik}$ on normalisointivakio.

  Lasketaan uudelleen merkitysten todennäköisyydet seuraavasti:

  $$P(s_k) = \frac{\sum_{i=1}^{I} h_{ik}}{\sum_{k=1}^{K} \sum_{i=1}^{I} h_{ik}} = \frac{\sum_{i=1}^{I} h_{ik}}{I}$$

Huomioita EM-algoritmista

- Algoritmi on herkkä alustukselle.
- $l(C|\mu)$ paranee joka kierroksella
- Mitä suurempi määrä luokkia (merkityksiä), sen parempi $l(C|\mu)$. Ongelmana on siis ylioppiminen (jos soveltaa sellaisenaan eri luokkamäärien vertailuun).

Paluu lähtökuoppiin

Joissain tilanteissa on OK epäonnistua.

Esim. sanaleikit, 'In AI, much of the I is in the beholder'
Vrt. 'Beauty is in the eye of the beholder' (kauneus on katsojan silmässä)

Hypoteesi (Kilgariff): On tavallista että useat merkityksistä yhtäaikaa läsnä:

'For better or for worse, this would bring competition to the licenced trade'
competition - competitors vs. competition - the act of competing

Mahdollinen selitys: ihmiset eivät disambiguoi sanoja vaan tulkitsevat lauseita ja tekstejä. Jos kaksi sananmerkitystä johtavat samaan lauseen tulkintaan, sananmerkityksiä ei ole tarpeen eritellä.

Disambiguointiongelmia esim.

- systemaattinen polysemia (tekemisen akti vs. osallistujat/yhteisö)
- pisteen merkitys (lauseen loppu vs. muu)
- erisnimi vai yleisnimi 'Brown', 'Bush'
- etu- vai sukunimi 'Pentti Jaakko'

Muita menetelmiä

Muita ohjatun oppimisen menetelmiä: kNN

- valitaan k lähintä luokiteltua esimerkkiä, ja luokitellaan tämä esimerkki enemmistöäänestyksellä.
- Sopii hyvin harvalle datalle
- Edellyttää ainoastaan 'samankaltaisuuden' määrittelyn + mittauksen lähimpiin samankaltaisiin (kompleksisuus O(Nd) jossa N datan määrä ja d dimensio)

Muita ohjaamattoman oppimisen menetelmiä

Klusterointimenetelmiä:
K-means (Schütze soveltanut merkitysten ryhmittelyyn), SOM, hierarkkiset klusterointimenetelmät, erilaiset samankaltaisuusmitat

Menetelmien vertailua

- Senseval-projekti: laaja WSD-menetelmien vertailu yhteisellä datalla
- aineisto + evaluoinnit webissä

Kriittinen kommentti lähtökohtaoletuksesta

Alussa määriteltiin ongelma tähän tapaan: oletetaan sana w, jolle olemassa k erillistä merkitystä $s_1 \ldots s_k$.

Kuitenkin on kyseenalaista, voidaanko sanojen eri merkityksiä jäännöksettömäs tarkastella "pistemäisinä", diskreetteinä oliona.