May 25, 2006

# Peidetud Markovi mudel

HMM on lõplik automaat, milles olekute siiretel on tõenäosused ja mis väljastab sõnade järjendi.
Eelmisele valemile vastab HMM, mille olekuteks on märgendid, tõenäosus minekuks ühest olekust teise on $p(t_i|p_{i-1})$ ja olekus $t_i$ stringi väljastamisel on $p(w_i|t_i)$.
Kui on antud sõnade järjend, leida olekute järjend, mida masin läbib, et väljastada see sõnade järjend, et tõenäosus oleks suurim.
Kust need tõenäosused sinna tulid?

# HMM-i treenimine

Baum-Welchi algoritm
Kui me teame algseid väärtusi, siis me saame leida mudeli olekute kombinatsioonide sagedused ja nende põhjal tõenäosused ümber hinnata.

1. Määra algsed tõenäosused
2. Rakenda BW algoritmi algsetel andmetel ja arvuta uued tõenäosused.
3. korda sammu 2 kuni lõpetamise kriteerium on täidetud (tõenäosused muutusid vähem kui mingi lävi vms)

# Dekodeerimine

Kui mudel on valmis, saame leida väljundsõnade jada tõenäosuse või olekute jada, mis selle genereerisid.
Dekodeerimine - leida kõige suurema tõenäosusega olekute jada.

**Viterbi algoritm**

Kui me vaatame olekut S ajahetkel t, siis seda olekut läbiv parim tee on olekuni S viiva parima tee ja olekust S lõppolekusse viita tee konkatenatsioon (tõenäosuste korrutis).
Kui me teame parimaid teid ajahetkel t-1, siis on lihtne leida parimaid teid ajahetkel t: tuleb vaadata kõigi teede võimalikke jätke ning leida neist parimad iga oleku jaoks.
Tee konstrueerimist alustatakse ajahetkest t=1.
Tee, mis annab lõppolekus suurima tõenäosuse, on parim.
Kui automaadis leidub tippe, millel unikaalne märgend ja nende vahel on ainult üks tee, siis leitakse parim tee kummagi poole jaoks eraldi.

# Bi- ja trigrammidest

Optimaalne märgendite järjend $\tilde{T}$ on see, mille korral $P(T|W)$ on suurim.
$P(T|W) = P(T) * P(W|T)/P(W)$ - vaja leida selle maksimumi
$P(W)$ on antud - vaja leida maksimum $(PT) * P(W|T)$
Kumbagi ei saa otse arvutada - vaja leida lähend

$$P(T) = P(t_1) * \prod_{i=2}^{L} P(t_i|t_1, ..., t_{i-1})$$

$P(t_i|t_1, ..., t_{i-1})$ on liiga palju, vaja lähendada
$P(t_i|t_1, ..., t_{i-1}) \approx P(t_i|P_{i-N+1}, ...t_{i-1})$
N = 2 - bigramm, N = 3 - trigramm
$P(W|T) = \prod_{i=1}^{L} P(w_i|t_i)$

# Smoothing

90% võimalikest trigrammidest ei esine isegi hiigelsuurtes korpustes. Kui meil on harvaesinev trigramm, siis tema väärtuse silendamiseks kasutatakse valemit: $P(t_i|t_{i-2}, t_{i-1}) =$
$\lambda_3 * P_3(t_i|t_{i-2}, t_{i-1}) + \lambda_2 * P_2(t_i|t_{i-1}) + \lambda_1 * P_1(t_i) + \lambda_0 * P_0$
$\lambda 0 + \lambda_1 + \lambda_2 + \lambda_3 = 1$

# Mudeli suurusest

V - sõnastiku suurus (min 100.000)
N - mudeli suurus (1,2,3)
S - märgendite arv
Sõnapõhine mudel: *parameetrite arv* $\sim V^N$
Sõnaklassipõhine mudel: *parameetrite arv* $\sim S^N + S * V$

# Eesti keele morf. ühestamine HMMiga

vt (Kaalep&Vaino 1998) Märgenditest
ÜM valimisel lähtutakse järgmistest nõudmistest.

1. ÜM peavad esindama süntaktiliselt selgelt eristuvaid klasse, s.t nad peavad oma kontekstis olema selgelt eristatavad.
2. ÜM klassid peaksid olema küllalt suured, et statistika nende kontekstide suhtes oleks usaldusväärne. ÜM klasside suuruse nõue on samaväärne nõudega, et ÜM klasside arv oleks teatud piirides. Paljude keelte puhul peetakse õigustatuks ÜM arvu alla 100, ehkki näiteks rootsi keele puhul kasutatakse 180 märgendit
3. ÜM tuleks valida nii, et ühestamisest oleks ikka tõesti kasu. Nt. tuleks panna nimetavas ja omastavas käändes sõnad eri klassidesse, sest nad on sageli vormilt homonüümsed ja neid saabki eristada ainult konteksti alusel.

# Eesti HMM jätkub

Praegu kasutame 88 ÜM, mis on valitud järgmiselt. Eristatakse omadussõnu, põhiarvsõnu, järgarvsõnu, nimisõnu, pärisnimesid, isikulisi asesõnu, muid asesõnu, lühendeid, verbe, alistavaid ja rinnastavaid sidesõnu, hüüdsõnu, ees- ja tagasõnu, määrsõnu, punktuatsioonisümboleid ja tundmatuid sõnu.
Käändsõnade puhul eristatakse 5 käänet: nimetavat, omastavat , osastavat , lühikest sisseütlevat e. aditiivi ja kõiki muid. Isikuliste asesõnade puhul eristatakse lisaks ka kolme isikut. Ei eristata ainsust ja mitmust.
Verbide puhul eristatakse kokku 13 ÜM-i: ei, ära, esimene pööre, teine pööre, kolmas pööre, kaudne kõneviis, pole ja polnud, da-infinitiiv, 0-lõpuline vorm, tingiva kõneviisi vormid, käskiva kõneviisi vormid, ma-infinitiivi vormid, partitsiibid. Ei eristata ainsust ja mitmust ega aega.

# Eesti HMM treenimine

Treeningkorpuseks oli G. Orwelli 1984 eestikeelne tõlge, v.a. Lisa, kokku 75 000 sõna. Kogu tekst oli morfoloogiliselt analüüsitud, seejärel ühestatud T. Puolakaise piirangut grammatikal põhineva ühestajaga (mis jättis 16% sõnadest mitmeseks, kuid ühestas ligi 100%lise korrektsusega) ja lõpuks käsitsi kontrollitud ning ühestatud.
Treenimisel oli kaks faasi: esialgsete tabelite koostamine ja tabelite parandamine treeningkorpuse põhjal. Treenimiseks kasutasime ISSCO ühestajat, mis on vabalt kasutatav tarkvara ja mille saime ISSCO koduleheküljelt http://issco.unige.ch.

# Eesti HMMi treenimine

Tabelite koostamiseks sai ühestaja 4 sisendit:

1. Morfoloogiliselt analüüsitud, kuid ühestamata tekst
2. Morfoloogiliselt analüüsitud ja ühestatud tekst
3. Ühestamisel kasutatavate märgendite (ÜM) loend
4. Teisendustabel morfoloogilistelt märgenditelt ÜM-idele

Ühestatud teksti põhjal arvutati:

1. Tõenäosuste vektor $E = \{e_1 e_2 ? e_w\}$, kus $e_i$ on tõenäosus, et $m_i$ on lauses esimene morfoloogiline märgend.
2. Maatriks $P = \{p_{kl}\}$, kus $p_{kl}$ on tõenäosus, et märgendile $m_k$ eelneb märgend $m_l$.

Vaadeldes korraga nii ühestatud kui ühestamata teksti, koostati maatriks $X = \{x_{kl}\}$, kus $x_{kl}$ on tõenäosus, et mitmesusklassi $v_l$ kuuluvatest märgenditest tuleb valida märgend $m_k$

# Tulemused

Pärast treeningkorpuse automaatset ühestamist võrdlesime käsitsi ja automaatselt ühestatud tekste. Selgus, et 12,67 % sõnadest olid valede ÜMidega. Seejuures kõige sagedasemad vead olid nud- või tud-partitsiibi määramine verbi asemel omadussõnaks (24 % kõigist vigadest) ja ei määramine verbi asemel määrsõnaks (13%).
Käsitsi tabeli parandamisel saadi 7.1%

# Ülevaade

ML on AI haru, mis uurib algoritme, mis suudavad õppida kogemuse põhjal või olemasolevate teadmiste reorganiseerimise teel. MLsüsteem koosneb enamasti kahest osast:

- soorituskomponent
- õppimiskomponent

Kogemus - näidete hulk.
Õppimiskomponent otsib näidete baasil optimaalse esituse. Et otsing on ajaliselt keerukas, kasutatakse heuristikaid.

Antud: näide sisendi ja näide väljundi kohta.
Ülesanne: leida seosed ja kasutada neid tundmatule sisendile väljundi genereerimiseks
MI algoritm klassifitseerib uued sisendmustrid sobivatesse väljundkategooriatesse - lause kujutatakse morfosüntaktiliste märgendite järjendiks.
Muster: sõna, mis on fookuses ja selle kontekst.
Kujutis: märgend ja kontekst. Kontekst võib olla 1 sõna kummalgi pool (trigram) kuni terve lause.

# Ülevaade masinõppimise meetoditest

- Tabelist vaatamine
- Juhtumipõhine õppimine
- Reeglite ja otsustuspuude genereerimine
- Konnektsionism, närvivõrgud.

ahne õppimine - teadmised hangitakse kohe, kui näidet nähakse
laisk õppimine - õpitakse siis kui on vaja

# Juhtumipõhine õppimine

Näited esitatakse kui tunnuste väärtuste vektorid. Kui sooritusKomponendis leitakse uus muster (vektor), mida baasis pole, kasutatakse sarnasusfunktsioone kõige sarnasema vektori leidmiseks.
Sarnasus: kattuvusprotsent, tunnuste olulisus, väärtuste kaugusfunktsioon. Näidetele kaalude andmine.
Memory Based Tagging - 96.4

# Otsustuspuu genereerimine

Sarnasused näidete vahel võimaldavad automaatselt luua otsustuspuid ja uue mustri jaosk sobivat puud leida.
Otsustuspuu on andmestruktuur, kus tipud esitavad teste ja kaared vastuseid. Probleem on lahendatud, kui juurtipust leidub tee mõne leheni.
Õppimine toimub sel teel, et näidete hulk jaotatakse alamhulkadeks selle järgi, kas näidetel on ühiseid tunnuseid, kuni kõik alamhulgad on homogeensed.
Oluliste tunnuste õige valik.
NP-täielik ülesanne.
Kasutatakse ka fraasistruktuuripuude korral.
Kõige tõenäolisem otsuste järjekord annab kõie tõenäolisema väljundi.
Pilt.
Mudeli suurus sõltub näidete arvust. Valib ise sobiva kontekstisuuruse.
Võib kasutada ka suuremaid kontekste.
Paremini kasutatav kaugete sõltuvuste korral. 96.4-96.5%

# Eesti keele TreeTagger

Märgenditeks MA sõnaligid (13) Treeningkorpus üle 300.000 sõna. Testkorpus 60.000 sõna. Vead (2400 sõna): V>S (13), V>S(12),A>S, S>A,K>D (10),O>V, S>V(7),N>P,K>S(3), S>K, D>A, J>D, G>S (2) Tulemus: 94,22% sõnesid

# Närvivõrgud

Joonis.
Vaatas 3 sõna vasakule, 2 paremale - võrdväärsed tulemused trigram-meetodiga. Tundmatuid mustreid analüüsib paremini. Vähem parameetreid - saab kasutada laiemat konteksti kui trigram-mudel.

# Statistiline parsimine

Morfosüntaktiline märgendamine eeldas käsitsi märgendatud korpust.
Parsimine eeldab puudepanka.
Programmi treenitakse puudepangal ja testitakse siis uuel tekstil, tulemust võrreldakse inimese poolt loodud puuga.
Saagis= leitud õiged moodustajad/kõik õiged moodustajad
Täpsus= leitud õiged moodustajad/kõik leitud moodustajad.
Moodustaja on õige, kui ta algab õigest kohast, lõppeb õiges kohas ja ta märgend on õige.
Keskmistel statistilistel parseritel on saagis/täpsus 75%, kui kasutavad leksikaalset infot, siis isegi kuni 88%.

# Näide

```
(s (np (det The) (noun stranger))
   (vp (verb ate)
       (np (det the) (noun doughnut))
       (pp (prep with) (np (det a) (noun fork)))))

\begin{verbatim}
(s (np (det The) (noun stranger))
   (vp (verb ate)
       (np (det the) (noun doughnut)
       (pp (prep with) (np (det a) (noun fork))))))
```

# Statistiline parsimine

- Tuleb leida võimalikud analüüsid
- Leida nende tõenäosused
- Valida neist suurima tõenäosusega analüüs

# PCFG

| | | | |
|---|---|---|---|
| s | $\longrightarrow$ | np vp | (1.0) |
| vp | $\longrightarrow$ | verb np | (0.8) |
| vp | $\longrightarrow$ | verb np np | (0.2) |
| np | $\longrightarrow$ | det noun | (0.5) |
| np | $\longrightarrow$ | noun | (0.3) |
| np | $\longrightarrow$ | det noun noun | (0.15) |
| np | $\longrightarrow$ | np np | (0.05) |

"The salespeople sold the dog biscuits"

# Analüüsi tõenäosus

s - lause

$\pi$ - konkreetne analüüs

c - moodustaja

r(c) - reegel, mille vasak pool on c

$$p(s, \pi) = \prod_{c} p(r(c))$$

$$1.0 * 0.3 * 0.8 * 0.15 = 0.036$$

# Tõenäosuste saamine

1. Muuta olemasolev grammatika PCFG-ks ehk leida reeglite tõenäosused
2. Rakendada grammatikat tekstile e leida parser ja käivitada see
3. Leida analüüsid, millel on suurim tõenäosus

Tõenäosused omistatakse reeglite loendades, kui palju neid treeningkorpuses kasutati ja mitut sama vasaku poolega reeglit üldse kasutati.
Seda liiki grammatikate saagis/täpsus on 75%.
Ei saa hakkama, kui grammatikas puudub reegel.
Markovi grammatikad genereerivad ise reegli (moodi tabeli).
Tõenäosus, et det adj noun annab np, on suur, samas tõenäosus, et prep annab np on minimaalne.

$$p(r|l) = \prod_{t_i \in r} p(t_i|l, t_{i-1})$$