Johdatus tekoälyyn, kevät 2017, Teemu Roos **Harjoitus 3, palautus 9.-10.2.2017**

Tehtävä 1. Naivi-Bayes ja SPAM (2 pistettä).

Naivi-Bayes-mallia käytetään roskapostin suodattamiseen. Mallin todennäköisyysparametrien oppimista varten on kerätty valmiiksi luokiteltuja sähköpostiviestejä. Seuraavassa on joidenkin sanojen esiintymismäärät kummassakin luokassa (spam ja ham):

sana	**spam**	**ham**
million	156	98
dollars	29	119
adclick	51	0
conferences	0	12
yht.	95 791	306 438

Arvioi ehdolliset todennäköisyydet $\mathrm{P}(\mathrm{SANA}_i = s \mid \mathrm{spam})$ ja $\mathrm{P}(\mathrm{SANA}_i = s \mid \mathrm{ham})$, missä $s \in \{\text{million, dollars, adclick, conferences}\}$.

Laske näitä hyödyntäen seuraavat todennäköisyydet:

i. (0.5 pistettä) $\mathrm{P}(\mathrm{SANA}_i \neq \text{million})$
eli todennäköisyys, että satunnaisesti valittu sana *ei* ole *million*, kun ei tiedetä, onko viesti spammia vai hammia

ii. (0.5 pistettä) $\mathrm{P}(\mathrm{spam} \mid \text{million})$
eli todennäköisyys, että viesti on spammia, kun ensimmäinen sana on *million*

iii. (1 piste) $\mathrm{P}(\mathrm{spam} \mid \text{million}, \text{dollars}, \text{adclick}, \text{conferences})$

Käytä prioritodennäköisyytenä arvoa $\mathrm{P}(\mathrm{spam}) = 0.5$.

Vinkkejä: Kohdassa i muista marginalisointikaava ja ärsyttävä nimittäjä. Kohdassa ii muista Bayesin kaava. Pienten todennäköisyyksien kohdalla on syytä käyttää jotakin alarajaa, esim. 0.00001. Kohdassa iii muista että jos saat laskettua osamäärän

$$\mathit{Odds} = P(\text{spam} \mid \text{evidenssi}) / P(\text{ham} \mid \text{evidenssi}),$$

saat todennäköisyyden kaavalla $P(\text{spam} \mid \text{evidenssi}) = \mathit{Odds}/(1 + \mathit{Odds})$.

Tehtävä 2. Roskapostisuodattimen toteutus (2 pistettä).

Lataa kurssin sivulta tiedostot `spamcount.txt` ja `hamcount.txt`, jotka sisältävät valmiiksi lasketut sanojen esiintymismäärät SpamAssassin-kehittäjien tarjoamasta sähköpostikorpuksesta.[1] Yleisimmin esiintyvät epäkiinnostavat sanat kuten *the, to, a, ...*, sanat jotka esiintyvät vain kerran ja muuta roskaa on jätetty huomiotta. Tiedostot on järjestetty siten, että useimmin esiintyneet sanat tulevat ensin:

```
top-10 spam-sanat:        top-10 ham-sanat:
 624 free                 1776 list
 465 email                1263 lists
 414 money                1204 use
 410 please               1007 exmh
 410 mail                  987 like
 383 list                  952 some
 360 click                 919 wrote
 358 content               909 linux
 339 business              895 listinfo
 306 information           893 rpm
```

Spam-tiedostossa esiintyy 6 245 eri sanaa ja niiden esiintymien yhteismäärä on 75 268. Ham-tiedostossa esiintyy 16 207 eri sanaa ja niiden esiintymien yhteismäärä on 290 673.

Toteuta roskapostisuodatin, joka lukee ko. tiedostot, laskee niiden perusteella todennäköisyysarviot $\mathrm{P}(\mathrm{SANA}_i = s \mid \mathrm{spam})$ ja $\mathrm{P}(\mathrm{SANA}_i = s \mid \mathrm{ham})$ tiedostoissa esiintyville sanoille s. Sen jälkeen suodatin lukee uuden sähköpostiviestin ja laskee todennäköisyyden, että se on spammia.

Testaa suodatintasi kurssin sivulta löytyvillä esimerkkiviesteillä, joista on jo valmiiksi poistettu välimerkit ja korvattu isot kirjaimet pienillä.

Vinkkejä: Testiviestissä saattaa esiintyy sanoja, joita ei ole esiintynyt opetusaineisossa. Näiden kohdalla kannattaa taas käyttää jotakin nollasta poikkeavaa todennäköisyyttä. Numeeristen epätarkkuuksien välttämiseksi kannattanee laskea osamäärän $Odds$ asemesta sen logaritmia

$$logOdds = logOdds + \log(\mathrm{P}(\mathrm{SANA}_i = s \mid \mathrm{spam})/\mathrm{P}(\mathrm{SANA}_i = s \mid \mathrm{ham})).$$

[1] `spamassassin.apache.org/publiccorpus`

Tehtävä 3. Shakki. (2 pistettä)

a) (1 piste). Toteuta valmiina annettuun runkoon heuristinen evaluointifunktio (luokkaan `YourEvaluator`), joka arvioi shakkilaudan tilanteen. Funktion tulee palauttaa sitä suurempi arvo, mitä luultavammin peli päättyy valkoisen voittoon.

Arviointifunktio voi riippua mm. kummankin väristen nappuloiden määrästä laudalla, nappuloiden sijoittumisesta laudan tärkeisiin kohtiin, jne. Testaa arviointifunktiota peluuttamalla siihen perustuvaa shakkialgoritmia sellaista algoritmia vastaan, joka perustuu valmiina tarjottuun evaluointifunktioon (luokka `OurEvaluator`).

Netbeans-ohje: Avaa maven-projekti Netbeansissa ja paina `Clean&Build`-nappulaa. Projektin pitäisi kääntyä ja toimia. Parantele `YourEvaluator`-luokan evaluointifunktiota ja kokeile, pystytkö voittamaan `OurEvaluator`-funktioon perustuvan pelaajan (a.k.a "Deep Glue").

b) (1 piste). Kokeile kuinka hyvin oma shakkibottisi pärjää turnauksessa. Lataa oma shakkitekoälysi turnauspalvelimelle (osoite ilmoitetaan myöhemmin) ja laita se pelaamaan toisten opiskelijoiden tekemiä shakkitekoälyjä vastaan.

HUOM: Shakkiturnaus jatkuu useamman viikon, joten älä käytä vielä tässä vaiheessa liikaa aikaa shakkibottisi viilaamiseen muiden tehtävien kustannuksella.

Huom: Älä aja shakkibotteja melkissä tai muissa interaktiivisissa servereissä. Ukko-klusterin solmuja, joita ei ole varattu muihin tarkoituksiin, voi käyttää testeihin. Voit kirjautua niihin ssh:lla, esim. `$ ssh ukko***.hpc.cs.helsinki.fi`, missä `***` on solmun numero. Solmujen kuormitusta ja varauksia voi tarkkailla osoitteessa `http://www.cs.helsinki.fi/ukko/hpc-report.txt` . Huomaa esim. vähimmällä kuormalla olevien solmujen lista rivillä 5. "Needs restart"-huomautuksesta ei kannata huolestua.[2]

[2] Koko kurssille ylimäärinen laskaripiste, jos saatte klusterin sulamaan, ks. `http://www.cs.helsinki.fi/en/story/63229/bringing-ukko-cluster-down`.