Tehtävä 1. Probabilistinen päättely (3 pistettä)

Tutustu luentokalvoissa esitettyy auton käynnistymistä kuvaavaan probabilistiseen malliin, jossa esiintyvät seuraavat tapahtumat/muuttujat:

$$
\begin{aligned}
A = 1, &\quad \text{joss akussa on virtaa,} \\
R = 1, &\quad \text{joss radio soi,} \\
S = 1, &\quad \text{joss sytytys toimii,} \\
B = 1, &\quad \text{joss tankissa on bensaa,} \\
K = 1, &\quad \text{joss moottori käynnistyy,} \\
L = 1, &\quad \text{joss auto liikkuu.}
\end{aligned}
$$

a) (1 piste).

- *i* Kuinka monta erilaista alkeistapahtumaa malli käsittää?
- *ii* Miten helppoa olisi määritellä niiden todennäköisyydet "lonkalta" luettelemalla?
- *iii* Kuinka monta alkeistapahtumaa on mallissa, joka sisältää N satunnaismuuttujaa, joista jokaisella on k mahdollista arvoa?
- *iv* Kuinka monta todennäköisyysarvoa jouduttiin määrittelemään, kun automalli esitettiin Bayes-verkkona? Huomaa, että yleisesti vastaus riippuu merkittävästi sekä solmujen että kaarien määrästä.
- *v* Kuinka monta todennäköisyysarvoa voitaisiin enimmillään joutua määrittelemään mallissa, jossa on $N = 4$ muuttujaa, joista jokaisella on k mahdollista arvoa? Muista, että verkossa ei saa olla syklejä. (Vapaaehtoinen lisätehtävä: Yleistä mille tahansa lukumäärälle $N \geq 1$. Avuksi voi olla geometrisen sarjan summakaava $1 + k + k^2 + \ldots + k^{N-1} = (k^N - 1)/(k - 1)$, kun $k \neq 1$.)

Tehtävä 1 jatkuu seuraavalla sivulla...

b) (1 piste). Toteuta algoritmi, joka generoi monikkoja (A, R, S, B, K, L). Generoi ensin muuttuja A siten, että se saa arvon 1 todennäköisyydellä 0.9. Generoi sen jälkeen muuttuja R siten, että se saa arvon 1 todennäköisyydellä 0.9, jos $A = 1$. Jos $A = 0$, R saa aina arvon 0. Jatka näin, noudattaen luentokalvoissa tai kurssimonisteessa annettuja todennäköisyyksiä, kunnes kaikki muuttujat on generoitu. Muista generoida solmut siinä järjestyksessä, että nuolen alkupäässä oleva solmu generoidaan ennen solmua, joka on nuolen loppupäässä.

Generoi näin $n = 100000$ monikon otos. Arvioi otoksen avulla seuraavia todennäköisyyksiä:

i) $P(A \mid R, B, \neg K)$.
ii) $P(K \mid R, S, B)$.
iii) $P(K \mid \neg R, S, B)$.

Esitä tulkinta saamillesi lukuarvoille. Vastaavatko ne intuitiota?

Kuvitellaan seuraavanlainen tilanne: Alueella jossa asut, tapahtuu maanjäristys todennäköisyydellä 0.009. Toisaalta (riippumatta maanjäristyksistä) asuntoosi tunkeutuu varas todennäköisyydellä 0.0032. Maanjäristyksen sattuessa varashälytin hälyttää todennäköisyydellä 0.81. Jos asuntoon murtaudutaan, hälyttää hälytin todennäköisyydellä 0.92. Jos käy niin, että maanjäristys ja murto tapahtuvat samalla kertaa, hälyttää hälytin todennäköisyydellä 0.97. Ilman maanjäristystä tai varasta, hälytin hälyttää todennäköisyydellä 0.0095.

c) (1 piste). Muokkaa b-kohdan algoritmia siten, että voit generoida yllä kuvatun mallin perusteella kolmikkoja M, V, H (maanjäristys, varas, hälytys). Generoi $N = 100000$ kokonaista kolmikkoa. Laske kuinka suuressa osassa kolmikkoja, joissa pätee $H = 1$, pätee myös $V = 1$. Laske myös kuinka suuressa osassa monikkoja, joissa pätee sekä $H = 1$ että $M = 1$, pätee lisäksi $V = 1$.

Esitä tulkinta havainnoillesi. Kumpi edellä mainituista osuuksista on suurempi? Osaatko sanoa mitä se merkitsee?

Toista koe muutaman kerran, jotta saat kuvan siitä, miten paljon tulokset vaihtelevat satunnaisesti. Miten otoskoon N kasvattaminen vaikuttaa?

Tehtävä 2. Numeroiden luokittelu neuroverkolla (2 pistettä).

Toteuta valmiiseen Java-ohjelmarunkoon, tai kokonaan alusta, perseptronialgoritmi numeroiden tunnistamiseen. Tiedosto `mnist-x.data` sisältää yhteensä 6000 kuvaa, yksi kuva jokaisella tiedoston rivillä, ja jokainen kuva on 28 x 28 pikseliä (eli jokaisella rivillä siis 28 x 28 = 784 arvoa). Jokainen pikseli on joko musta (-1) tai valkoinen (1). Tiedosto `mnist-y.data` sisältää näitä kuvia vastaavat luokat (0–9).

Käytä 5000 ensimmäistä esimerkkiä opetusaineistona ja 1000 viimeistä testiaineistona, jonka avulla voit arvioida luokitteluvirhettä.

a) Suorita Java-pohja kerran ja varmista, että projektin runkoon ilmestyy tiedosto `test100.bmp`, jossa on sata ensimmäistä numeroa suuruusjärjestyksessä. Tällä tavalla verifioidaan että datan lukeminen on onnistunut.

b) (1 piste) Muokkaa perseptronialgoritmia (`Perseptroni`-luokan `train()`-metodi) siten, että se oppii erottamaan numeroa ’3’ (`targetChar`) esittävät kuvat numeroa ’5’ (`oppositeChar`) esittävistä kuvista. (Luokkamuuttujan arvo on 1, kun kuva esittää kolmosta, ja -1 jos se esittää vitosta.) Minkä luokitteluvirheen saat aikaiseksi?

c) (1 piste) Kokeile eri numeroparien erottelua (muitakin kuin 3 vs 5). Mitkä numerot on helpoin erottaa toisistaan, mitkä vaikein?

Tehtävä 3. Numeroiden luokittelu lähimmän naapurin luokittimella (1 piste).

Korvaa edellisen tehtävän ohjelmassa perseptroniluokittelija lähimmän naapurin luokittelijalla (tai koodaa kokonaan alusta).

Luokittelija etsii siis kullekin testiaineiston esimerkille, X, sitä vastaavan lähimmän opetusaineiston esimerkin, X^{train}, ja palauttaa sitä vastaavan luokan Y^{train}. Huomaa, että toisin kuin perseptroniluokitinta, lähimmän naapurin luokitinta ei varsinaisesti tarvitse opettaa vaan kaikki työ tehdään luokitteluvaiheessa.

Testaa taas luokittelijaasi samoilla opetus- ja testiaineistoilla kuin edellisessä tehtävässä. Voit nyt ottaa mukaan kaikki luokat (numerot 0–9). Huomaa, että luokittelu 10:en luokkaan on vaikeampaa kuin luokittelu kahteen luokkaan, joten luokittelutarkkuus luultavasti alenee.