Tehtävä 1. Probabilistinen päättely (3 pistettä)

a) (1 piste).

i) Kuinka monta erilaista alkeistapahtumaa malli käsittää?
Vast.: Kukin kuudesta muuttujasta voi saada arvon 0 tai 1, joten kombinaatioita on $2^6 = 64$.

ii) Miten helppoa olisi määritellä niiden todennäköisyydet "lonkalta" luettelemalla?
Vast.: Työlästä ainakin. Ensinnäkin todennäköisyyksiä on melko paljon. Koska alkeistapahtumien todennäköisyyksien summa on yksi, riittää periaatteessa määritellä 63 arvoa ja viimeinen saadaan laskemalla $P(\omega_{64}) = 1 - P(\omega_1) - \ldots - P(\omega_{63})$. Toiseksi, vaikka tuntisi auton toiminnan kuin omat taskunsa, olisi vaikea sanoa suoraan, mikä on todennäköisyys $A = 1, R = 0, S = 1, B = 1, K = 1, L = 0$.

iii Kuinka monta alkeistapahtumaa on mallissa, joka sisältää N satunnaismuuttujaa, joista jokaisella on k mahdollista arvoa?
Vast.: Jos $N = 1$, on alkeistapahtumia selvästi k. Millä tahansa N yhden satunnaismuuttujan lisääminen malliin kasvattaa alkeistapahtumien määrän k-kertaiseksi. Siten alkeistapahtumia on k^N.

iv Kuinka monta todennäköisyysarvoa jouduttiin määrittelemään, kun automalli esitettiin Bayes-verkkona? Huomaa, että yleisesti vastaus riippuu merkittävästi sekä solmujen että kaarien määrästä.
Vast.: Luentokalvoissa lueteltiin 10 arvoa, joista $P(\text{"käynnistyy"} \mid \neg\text{"}S\text{"} \text{ tai } \neg\text{"}B\text{"}) = 0$ kattaa oikeastaan kolme eri tapausta: $(\neg\text{"}S\text{"}, \text{"}B\text{"})$, $(\text{"}S\text{"}, \neg\text{"}B\text{"})$ ja $(\neg\text{"}S\text{"}, \neg\text{"}B\text{"})$, joten tavallaan lukuarvoja, joista osa on 0, lueteltiin 12 kappaletta. Joka tapauksessa paljon vähemmän kuin 63 ja lisäksi niiden määrääminen oli helpompaa kuin yksittäisten alkeistapahtumien todennäköisyyksien määrääminen.

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 $N \geq 1$.)
Vast: Mitä enemmän kullakin muuttujalla on vanhempien ja sitä kautta vanhempien arvojen yhdistelmiä, sitä enemmän todennäköisyysarvoja joudutaan määrittelemään. Koska verkossa ei saa olla syklejä, on verkko, jossa on maksimaalinen määrä kaaria (eli vanhempia), sellainen, jossa yhdellä muuttujalla on nolla vanhempaa, yhdellä muuttujalla on yksi vanhempi, yhdella muuttujalla on kaksi vanhempaa, jne.
Sellaisen muuuttujan, jolla on r vanhempaa, kohdalla joudutaan määrittelemään k^r jakaumaa. Yhden k-arvoisen muuttujan jakauman määrittelyyn tarvitaan $k-1$ arvoa. Yhteensä muuttujan, jolla on r vanhempaa, kohdalla tarvitaan siis $k^r(k-1)$ arvoa. Näin koko verkossa todennäköisyyksiä joudutaan määrittelemään yhteensä $k^0(k-1) + k^1(k-1) + \ldots + k^{N-1}(k-1)$. Summaan voidaan soveltaa geometrisen sarjan summakaavaa

$$\sum_{r=0}^{N-1} k^r = \frac{k^N - 1}{k - 1},$$

millä saadaan arvojen lukumääräksi $k^N - 1$.
Kun $N = 4$ ja $k = 2$ (binaarimuuttujat), saadaan lukumääräksi $1 + 2 + 4 + 8 = 15$.
Tapauksessa $k = 6$ eli autotapauksessa, tulos on 63 eli sama kuin alkeistapahtumien kautta tarvittava!
Pointti tehtävässä oli huomata, että Bayes-verkon määrittelemiseen tarvitaan pahimmassakin tapauksessa korkeintaan yhtä monta lukuarvoa kuin alkeistapahtumien kautta ja yleensä merkittävästi vähemmän! Tästä lisää Todennäköisyysmallit-kurssilla.

b) (1 piste).

i) **Vast.:** $P(A \mid R, B, \neg K) \approx \dfrac{\text{Kuinka monessa } A, R, B, \neg K}{\text{Kuinka monessa } R, B, \neg K} = \dfrac{4640}{4640} = 100.0\%$

ii) **Vast.:** $P(K \mid R, S, B) \approx \dfrac{\text{Kuinka monessa } K, R, S, B}{\text{Kuinka monessa } R, S, B} = \dfrac{72412}{73155} \approx 99.0\%.$

iii) **Vast.:** $P(K \mid \neg R, S, B) \approx \dfrac{\text{Kuinka monessa } K, \neg R, \neg S, \neg B}{\text{Kuinka monessa } \neg R, \neg S, \neg B} = \dfrac{7915}{8013} = 98.8\%.$

Esitä tulkinta saamillesi lukuarvoille. Vastaavatko ne intuitiota?

Voidaan tehdä monta mielenkiintoista havaintoa. Ensinnäkin tilanteissa, joissa looginen kyllä/ei-tyylinen päättely antaa vastauksen, todennäköisyyspäättely antaa saman vastauksen. Kohdassa (i), on selvää, että akussa on virtaa, koska radio soi. Siksi todennäköisyys on 100 %.

Toiseksi, verkkoa tarkastelemalla, voidaan tehdä päätelmä, että radion soimisella ei ole merkitystä käynnistymisen kannalta, kunhan sytytys toimii ja bensaa on. Siksi ii- ja iii-kohtien tulos on itse asiassa sama, vaikka satunnaisvaihtelusta johtuva virhe tuottaakin pieniä eroja.

Edelleen koska ii- ja iii-kohdissa radion soimisella ei ole merkitystä eli K on ehdollisesti riippumaton satunnaismuuttujasta R annettuna S, B, on ko. kohtien tarkka arvo yhtä kuin $P(K \mid S, B) = 0.99$, mikä oli annettu mallia määritellessä.

Käytetään satunnaismuuttujille päivitetyn tehtävänannon mukaisia nimiä (M)aanjäristys, (V)aras ja (H)älytys.

c) (1 piste)
Vast.: $N_{H=1} = 1967$ ja $N_{H=1,V=1} = 263$. Tästä voidaan päätellä, että

$$P(\text{varas} \mid \text{hälytys}) \approx \frac{263}{1967} \approx 13.4\%.$$

Kuinka suuressa osassa monikkoja, joissa pätee sekä $H = 1$ että $M = 1$, pätee lisäksi $V = 1$?
Vast.: $N_{H=1,M=1} = 738$ ja $N_{H=1,M=1,V=1} = 2$, joista voidaan päätellä

$$P(\text{varas} \mid \text{hälytys ja maanjäristys}) \approx \frac{2}{738} \approx 0.3\%.$$

Esitä tulkinta havainnoillesi. Kumpi edellä mainituista osuuksista on suurempi? Osaatko sanoa mitä se merkitsee?
Vast.: Ensin laskettu tn on paljon suurempi. Jos siis saadaan tietää, että hälytin on lauennut, on syytä epäillä varkaan olevan murtopuuhissa. Jos kuitenkin saadaan lisäksi tietää, että alueella on sattunut maanjäristys, voidaan melko turvallisin mielin päätellä, että se onkin luultavasti laukaissut hälyttimen. Huomaa, että asuntomurron prioritodennäköisyys on 0.32%, joten posterioritodennäköisyys 0.3% on samaa luokkaa

sen kanssa. Toisin sanoen, hälytys maanjäristyksen sattuessa ei merkittävästi vaikuta varkauden todennäköisyyteen.
Tässä esimerkissä todennäköisyyslaskenta tuotti siis taas intuitiivisesti järkevän päättelytuloksen. Aina näin ei käy. Kuten luennolla oli puhetta, esimerkiksi tilanteissa, jossa Bayesin kaavaa voi soveltaa, antaa ihmisten intuitio usein kovin erilaisen (väärän) vastauksen kuin kaavaa soveltamalla saadaan.
Toista koe muutaman kerran, jotta saat kuvan siitä, miten paljon tulokset vaihtelevat satunnaisesti. Miten otoskoon N kasvattaminen vaikuttaa?
Vast.: Toistettaessa otoskoolla $n = 100\,000$ ensin laskettu todennäköisyysarvio heittelee noin noin prosentin suuntaan tai toiseen ja jälkimmäinen (pienempi) arvio noin kolme prosentin kymmetystä. Pienemmällä otoskoolla $n = 10\,000$ saadaan ensimmäiseksi todennäköisyydeksi esimerkiksi arviot $10.6\%, 14.8\%, 16.6\%, 11.0\%, 16.4\%$ ja toideksi arviot $0.0\%, 0.0\%, 1.4\%, 1.2\%, 0.0\%$. Isolla otoksella $n = 10^7$ saadaan tyypillisiä lukuja väliltä $14.9 \pm 0.2\%$ ja $0.3 \pm 0.1\%$. Haittapuolena on hitaus.
Esimerkkikoodia pythonilla löytyy kurssin sivulta.

Tehtävä 2. Numeroiden luokittelu neuroverkolla (2 pistettä).

Esimerkkitoteutus kurssin kotisivulla.

a) Ks. kuva 1. (Siinä kirjaimet on sijoitettu neliön muotoon, jotta kuva mahtuu paremmin sivulle.)

b) Luokitteluvirhe riippuu opetusaskeleiden määrästä. Alle kymmenellä opetusaskeleella virhe on luokkaa 50 %. Sadalla opetusaskeleella se vaihtelee välillä 5-50 % ja tuhannella opetusaskeleella virhe on luokkaa 5-15 %. Kun opetusaskeleiden määrä kasvaa vielä suuremmaksi, putoaa virhe tyypillisesti alle 5 %.

c) Ainakin numerot 4 ja 5 on keskimääräistä vaikeampi erottaa toisistaan. Virhe putoaa harvoin alle 3.5 %. Ehkä hieman yllättäen, 0 ja 8 on sen sijaan helppo erottaa toisistaan – ehkä kuvan keskellä olevien pikseleiden perusteella.

Kuva 1: Tehtävän 2a tulostus.

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

Lähimmän naapurin luokittimen esimerkkitoteutus kurssin sivulla (vähemmän kuvaavasti nimellä `MakeBMP.java`). Sen suoritetaan komennolla

```
java MakeBMP mnist-x.data mnist-y.data
```

Ensimmäinen huomio lähimmän naapurin luokittimesta on se, että se on todella hidas. Hitaus johtuu siitä, että jokaisen testiesimerkin kohdalla käydään läpi kaikki 5000 opetusesimerkkiä. Luokitin tulostaa jokaisen testiesimerkin kohdalla luokituksen (`result`) ja oikean luokan (`true`) sekä siihen astisen keskimääräisen luokitusvirheen (`error rate`). Lopputulos on 7.8 %. Huomaa, että tulosta ei voi suoraan verrata perseptronin tulokseen, koska lähimmän naapurin luokitin pystyy erottamaan kaikki numerot toisitaan, ei vain kahta vaihtoehtoa.

Esimerkkitoteutuksessa on myös k-lähimmän naapurin vaihtoehto, joka suoritetaan perusversion ($k = 1$) jälkeen. Hieman yllättäen isommat k:n arvot johtavat huonompaa luokittelutarkkuuteen. Tämä johtuu ilmeisesti opetusaineiston pienehköstä koosta: seuraavaksi lähimmät naapurit ovat jo liiankin erilaisia testiesimerkistä, jotta ne auttaisivat luokittelussa. Isommilla aineistoilla $k > 1$ on yleensä parempi, mutta silloin on syytä käyttää tiettyjä nopeutuskikkoja, jotta suoritusaika ei kasva sietämättömäksi.