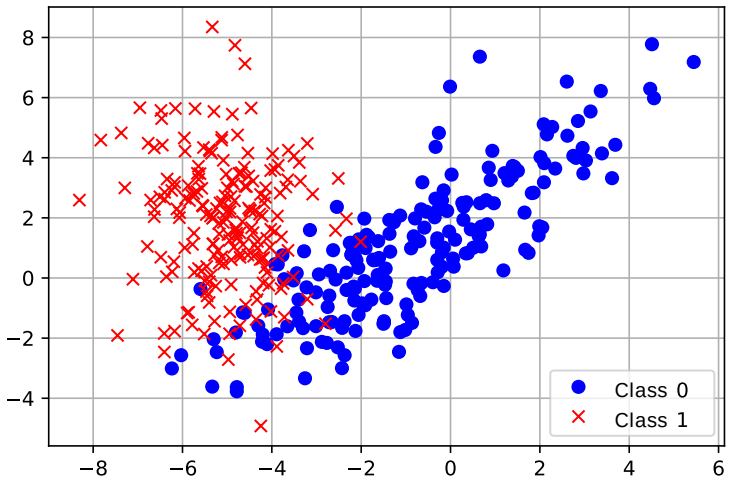
COMP.SGN.100 Signaalinkäsittelyn perusteet,
Harjoitus 12, 28.-30.4.2021

1. (*Kynä & paperi*) Suunniteltaessa lineaarista luokittelijaa kaksiulotteiselle datalle (kuva alla) saadaan opetusdatasta kahden luokan kovarianssimatriiseiksi ja keskiarvoiksi seuraavat:

$$\mathbf{C}_0 = \begin{pmatrix} 5 & 4 \\ 4 & 5 \end{pmatrix} \quad \mathbf{C}_1 = \begin{pmatrix} 1 & 0 \\ 0 & 4 \end{pmatrix}$$
$$\boldsymbol{\mu}_0 = \begin{pmatrix} -1 \\ 1 \end{pmatrix} \quad \boldsymbol{\mu}_1 = \begin{pmatrix} -5 \\ 2 \end{pmatrix}$$

   Laske projektiosuoran määräävä vektori $\mathbf{w}$. Älä käännä matriisia koneella, vaan käsin muistisäännöllä

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix}^{-1} = \frac{1}{ad - bc} \begin{pmatrix} d & -b \\ -c & a \end{pmatrix}.$$



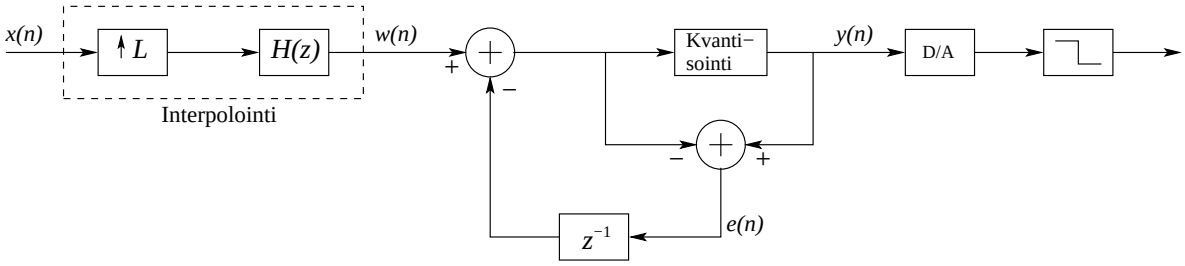
2. (*Kynä & paperi*) Jatkoa edelliseen tehtävään. Projektiosuoran suunnan lisäksi tarvitaan kohta, johon luokkien välinen raja vedetään. Yksinkertaisin tapa on sijoittaa se luokkien massakeskipisteiden puoliväliin. Käytännössä tämä tapahtuu projisoimalla data suoralle ja vertaamalla tulosta kynnysarvoon $c \in \mathbf{R}$:

$$\begin{cases} \text{Näyte } \mathbf{x} \text{ kuuluu luokkaan 0, jos } \mathbf{w} \cdot \mathbf{x} \geq c \\ \text{Näyte } \mathbf{x} \text{ kuuluu luokkaan 1, jos } \mathbf{w} \cdot \mathbf{x} < c \end{cases}$$

   Laske kynnysarvo c seuraavasti:

   - Laske mille arvolle $\boldsymbol{\mu}_0$ projisoituu.
   - Laske mille arvolle $\boldsymbol{\mu}_1$ projisoituu.
   - Luku c on näiden keskiarvo.

3. (*Matlab*) Toteutetaan Matlabilla ensimmäisen asteen kohinanmuokkain.



Lataa testisignaali `seiska.wav` kurssin Moodlesta (`Harj_12.zip`). Lataa se Matlabiin komennolla `audioread` muuttujaan `x` sekä näytteenottotaajuus muuttujaan `Fs`. Skaalaa signaalin lukuarvot kvantisointia varten välille $[-1, 1]$. Interpoloi signaali kertoimella 4 käyttäen valmista komentoa `interp`. Interpoloinnin tulos on lohkokaaviossa oleva signaali $w(n)$. Tätä signaalia lähdetään käsittelemään lohkokaavion mukaisesti. Alusta tätä varten muuttujat `e` ja `y` nolliksi: `e=zeros(1,length(w));`.

Lohkokaavion mukainen kohinanmuokkaus voidaan esittää pseudokoodina seuraavasti:

```
for n = 2 to length(w)
    z  = <w(n):n ja e(n-1):n erotus>
    y(n) = <z kvantisoituna pelkkään merkkibittiin>
    e(n) = <kvantisointivirhe>
end
```

Tulosta ruudulle ulostulosignaali `y` ja sen spektri. Tulosta spektri kurssin Moodlesta löytyvällä funktiolla `spektri.m` (`Harj_12.zip`).

Vertaa tulosta kvantisoimattoman signaalin `w` spektriin. Kuuntele lopuksi tulossignaali `y` (muista antaa `soundsc`-funktiolle oikea näytteenottotaajuus).

4. *(Matlab)* Muunna edellisen tehtävän kohinanmuokkaus toisen asteen versioksi, jossa ensimmäisen asteen järjestelmän $H(z) = 1 - z^{-1}$ tilalle tulee toisen asteen järjestelmä $H(z) = (1 - z^{-1})^2$. Tee vastaavat testit kuin edellisessä tehtävässä ja vertaa tulosta.

5. *(Matlab)* Tarkastellaan käsinkirjoitettujen numeromerkkien (0...9) tunnistusta LDA:lla. Lataa tiedosto `mnist.zip` kurssin Moodlesta (`Harj_12.zip`).

   Tiedosto sisältää seuraavat muuttujat:

   - `X_train`: 60000 opetusmerkkiä, joiden koko on 24×24 pikseliä.
   - `X_test`: 10000 testimerkkiä, joiden koko on 24×24 pikseliä.
   - `y_train`: opetusmerkkien luokkatieto (0...9).
   - `y_test`: testimerkkien luokkatieto (0...9).

   Opeta LDA-luokitin opetusdatalla, ja laske tarkkuus testidatalle. Tarvittavat vaiheet ovat seuraavat:

   (a) Piirrä yksi merkki ruudulle tarkistaaksesi miltä data näyttää: `imshow(squeeze(X_train(1, :, :)))`. Varmista myös että `y_train`-vektorin ensimmäinen arvo vastaa kuvaa.

   (b) Muunna 24×24-kokoiset bittikartat 576-ulotteisiksi vektoreiksi. Komento on `reshape`, ja tavoite muuntaa `X_train` koosta $60000 \times 24 \times 24$ kokoon 60000×576 (ja `X_test` vastaavasti).

   (c) Opeta LDA-luokittelija Matlabin valmiilla `fitcdiscr`-komennolla. Tässä voi kestää vähän aikaa.

   (d) Ennusta luokat testidatalle (komento `predict`).

   (e) Laske luokitteluvirhe; esim. `acc = mean(y_pred == y_test)`, missä `y_pred` on LDA-luokittelijan ennuste.

   (f) Etsi yksi luokitteluvirhe. Löydät kaikki väärin menneet indeksit `find`-komennolla: `find(y_pred ~= y_test);` ja tulosta yksi näistä kuten kohdassa (a). Tarkista myös miten malli tulkitsi kuvan (vektorissa `y_pred`).