linear classifiers (generalization of LDA)

- case $C=2$: $Y \in \{-1, +1\}$ $\quad \hat{Y}_i = \text{sign}(w X_i^T + b) = \text{sign}(\tau w X_i^T + \tau b)$ $\quad \tau > 0$

interpret as a feature reduction: - define new 1-D feature $z_i = w X_i^T + b$

- apply threshold classifier to z_i: $\hat{Y}_i = \begin{cases} +1 & z_i \geq 0 \\ -1 & z_i < 0 \end{cases}$

- z_i are weighted sum of the original features X_{ij} with weight w_j.

$w_j >> 0$: feature j votes for class $+1$

$w_j << 0$: — " — -1

$w_j \approx 0$: feature j is irrelevant

[remark: many neural networks do the same, but with a non-linear formula:

$z_i = f_\theta(X_i)$ ← trainable network parameters]

- case $C>2$: two main solutions: one-against-the-rest and all-pairs

- one-against-the-rest: train one linear equation per class ⇒ scores z_{ik}

$$z_{ik} = w_k X_i^T + b_k \qquad (\text{LDA})$$

- we decide for the class with highest score or "unknown" if no score is high:

$$\hat{Y}_i = \begin{cases} \text{"unknown"} & \text{if } \forall k: z_{ik} < 0 \quad (z_{ik} < \varepsilon) \\ \arg\max_k z_{ik} & \text{otherwise} \end{cases}$$

important: only works if the magnitudes of z_{ik} are comparable, automatically true for LDA, otherwise scale $\tilde{z}_{ik} = \tau_k w_k X_i^T + b_k \cdot \tau_k = \tilde{w}_k X_i^T + \tilde{b}_k$ so that $\|\tilde{w}_k\| = 1$

- all-pairs (apply this if the underlying classifier is unable to give comparable scores z_{ik} or if the classifier cannot handle $C > 2$)

for each pair (k, k') with $1 \le k < k' \le C$, train a 2-class classifier,

e.g. linear: $z_{kk'} = w_{kk'} \cdot x^T + b_{kk'}$

if $z_{kk'} \ge 0$ ⇒ one vote for class k, $z_{kk'} < 0$: one vote for class k'

$\hat{y}$ = class with most votes (or "unknown" if no class received signif. more votes than the others)

- modern extensions: more sophisticated voting patterns ≙ each classifier votes for multiple classes ≙ "coding matrix approach", more robust

(group classes into clusters, classifier votes for one cluster against another one, each classifier can use a different grouping (clusters can have overlap))

- <u>How to define good weights w and offsets b?</u>

- have already learned LDA: $w_k = \Sigma_w^{-1} \cdot \mu_k$ $\quad \Sigma_w = \frac{1}{N} \sum_{i=1}^{N} (x_i - \mu_{y_i})^T (x_i - \mu_{y_i})$

interpretation: approximate the feature distribution for each class by an ellipse centered at mean μ_k, but with equal shape for all classes

↑ class mean $\quad$ ↑ within-class covariance $\quad \mu_k = \frac{1}{N_k} \sum_{i: y_i = k} x_i$

≙ fit the likelihood $p(x \mid y = k)$ by a multivariate Gaussian $N(\mu_k, \Sigma_w)$

- Q: are there other criteria to find good w?

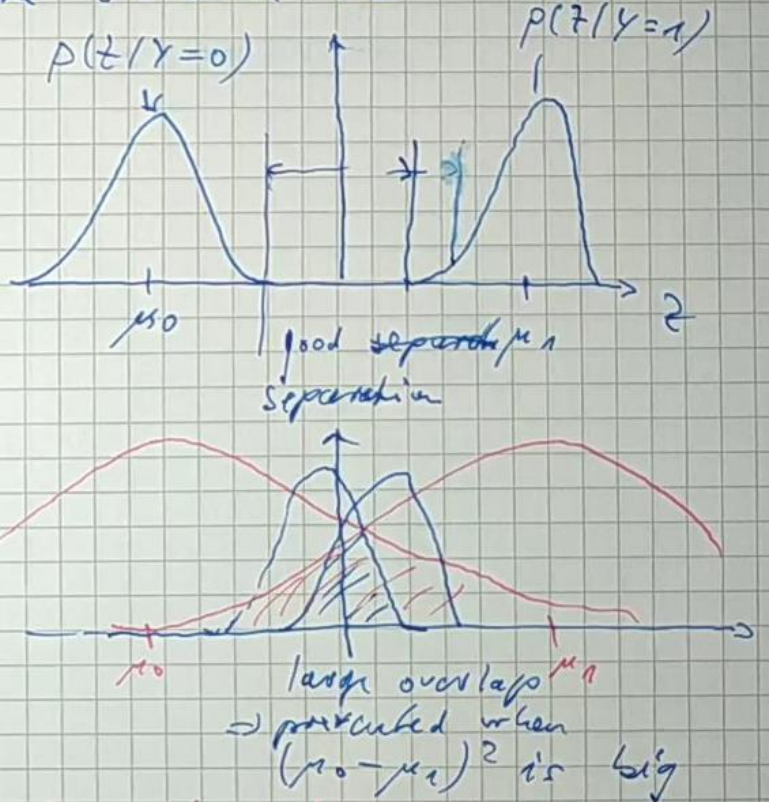
- Fisher's criterion ($C=2$) find a 1-D feature $z = w x^T + b$ such that the classes are well seperated in z
  - means μ_0 and μ_1 should have big distance

$$\hat{w} = \arg\max_w \frac{(\mu_0 - \mu_1)^2}{\sigma_0^2 + \sigma_1^2}$$

(σ_0^2, σ_1^2: variances of class 0 and 1 in z)

compute the analytical solution for $\hat{w}$

surprise: same result as LDA



large overlap: large $(\mu_0 - \mu_1)^2$ is not enough, also need small variance

- least-squares criterion, $Y = \{-1, 1\}$
  - find w such that $z_i = w x_i + b$ has $z_i \approx 1$ if $Y_i = 1$, $z_i \approx -1$ if $Y_i = -1$

⇒ measure the difference between actual and desired value by squared difference: $\hat{b}, \hat{w} = \arg\min_{w,b} \frac{1}{N} \sum_{i=1}^{N} (w x_i^T + b - Y_i)^2$

surprise: if classes are balanced ($N_{-1} = N_1 = \frac{N}{2}$) ⇒ get same solution as LDA

proof of this fact ⇒ homework

- Q: are there criteria where we get a different solution than LDA?
  yes: perceptron, support vector machine (later), logistic regression