Linear Discriminant Analysis
(see Alpaydin ch. 5 and section 6.6)

- Key idea: For each class \( y \):
  - Estimate \( P(\mathbf{x} \mid y) \) with a Gaussian (using shared covariance matrix)
  - Estimate prior \( P(y) \) as fraction of training data with class \( y \)
- Predict the \( y \) maximizing \( P(y) \cdot P(\mathbf{x} \mid y) \),
  Or maximizing \( \log(P(y)) + \log(P(\mathbf{x} \mid y)) \)
- What is the hypothesis class?

Generative model:

- Hypothesis class associated with set of distributions (e.g. gaussians)
- Each hypothesis associates a distribution on \( X \) with each class (or label) (distributions may be restricted - e.g. have same variance)
- Put "prior" \( P(y) \) on classes/labels
- To generate sample:
  - Pick label \( y \) from \( P(y) \)
  - Pick features \( x \) from \( P(x \mid y) \)

Exercise (OLD):

- Implement LDA for two classes and train on your iris2.arff data
- Compare the classifier you get from the one from Weka by:
  - Select "classify" tab, then "choose" button, then "Classification via Regression" inside the "meta" folder. Click on the ClassificationViaRegression (beside the choose button), in the pop-up box click choose and select "linear regression" from the functions folder. Click on "linear regression" (next to the choose button in the pop-up window) and change "attribute selection" to "no selection", "eliminate co-linear" to "false" and set the "ridge" parameter to 0.

Decision boundary

For 2-classes 0 and 1, predict on \( \mathbf{x} \)

- Predict \( y=1 \) if
  \[ P(y=1) \cdot P(\mathbf{x} \mid y=1) > P(y=0) \cdot P(\mathbf{x} \mid y=0) \]
- Predict \( y=0 \) otherwise
- Decision Boundary where
  \[ P(y=1) \cdot P(\mathbf{x} \mid y=1) = P(y=0) \cdot P(\mathbf{x} \mid y=0) \]
- Where is decision boundary?
  - Assume Gaussians with same sigma

\[ P(1)P(\mathbf{x} \mid 1) = P(0)P(\mathbf{x} \mid 0) \]
\[ \frac{G_1(\mathbf{x})}{G_0(\mathbf{x})} = \frac{P(0)}{P(1)} \]
\[ e^{-\frac{1}{2}stuff1} = \frac{P(0)}{P(1)} \]
\[ e^{-\frac{1}{2}stuff2} = \frac{P(0)}{P(1)} \]
\[ e^{-\frac{1}{2}stuff1 - stuff2} = \ln \left( \frac{P(0)}{P(1)} \right) \]
Linear threshold classification

- Linear threshold unit: predicts 1 if \( w \cdot x = w_0 \)
- LTU’s can represent:
  - Conjunctions like \((x_1 \text{ and } x_2 \text{ and not } x_3)\)
  - At least \(k\) of \(m\) functions like:
    - At-least-2-of \((x_1, x_3, \text{not } x_4)\)
- LTU’s can not represent
  - XOR functions
  - Complex disjunctions like \((x_1 \text{ and } x_2)\) or \((x_3 \text{ and } x_4)\)

LDA finds a informative projection

Other LTU algorithms:
- Perceptron algorithm
- Logistic regression
- Support vector machines