Supervised binary classification

Data Set \( \mathcal{D} = \sum (x_i, y_i) \sum_{i=1}^{N} \)

\( x_i \in \mathbb{R}^D \quad (5, 6, 0, 1, -18, ...) \)

\( y_i \in \{ -1, 1 \} \)

"Design Matrix"

\[
X = \begin{bmatrix}
    x_{i1} & x_{i2} & \ldots & x_{iD} \\
    \vdots & \vdots & \ddots & \vdots \\
    x_{N1} & x_{N2} & \ldots & x_{ND}
\end{bmatrix}
\]

Column is feature values across examples

Training

"discriminant function"

\[
f(x_i) \rightarrow y_i
\]

\{ Supervised learning is about finding f \}
Testing phase

\[ D_{\text{test}} = \sum_{i=1}^{3} x_{i} \sum_{i=1}^{3} \]

\[ X_{\text{test}} \]

\[ f(x_{i}) \]

\[ y_{1}, y_{2}, \ldots, y_{T} \]
**K-Nearest Neighbours**

**Training:**
- Store $X$ and $y$; choose $K$.

**Testing:**
- For $x_i \in X_{\text{test}}$
  - Find $K$ 'closest' examples $\exists x_1, x_2, \ldots, x_K$ from training set.
  - Look at corresponding $\exists x_{i1}, x_{i2}, \ldots, x_{iK}$
  - Choose majority
### Issues
- Ties
- Weight neighbours ('closest')

### Advantages
- only 1 choice
- simple, parallelizable
- training fast
- fast updates
- highly flexible/non-linear

### Disadvantages
- huge memory
- choose k
- class imbalance
- need to cover the space
- testing slow
- how to define 'distance'?
- uniform distance measure?