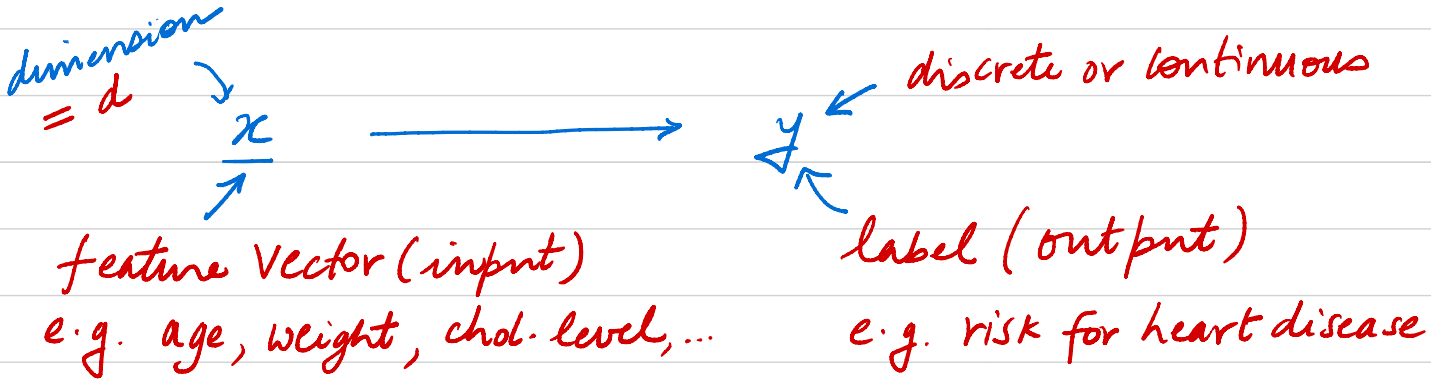
Machine Learning



Goal To learn mapping from $\underline{x}$ to y

Supervised Learning Use labelled training data

$$\tau = \{(\underline{x}_1, y_1), (\underline{x}_2, y_2), \ldots (\underline{x}_N, y_N)\}$$

Classification Label y takes on finite # values

If only two values $\{-1, 1\}$ – binary classification

If M values ($M > 2$) – M-ary classification

Classifier uses training data τ to form function f

f is used to predict labels for $\underline{x}$ outside τ

$$\hat{y} = f(\underline{x})$$

($\underline{x}$: feature vector $\notin \tau$)

Measuring Classifier Performance

Loss function $L(\hat{y}, y) = L(f(\underline{x}), y)$

0-1 Loss: $L(\hat{y}, y) = \begin{cases} 1 & \text{if } \hat{y} \neq y \\ 0 & \text{if } \hat{y} = y \end{cases}$

$$\tau = \{(\underline{x}_1, y_1), (\underline{x}_2, y_2), \cdots (\underline{x}_N, y_N)\}$$

Training Error

$$Err_{train}(\tau) = \frac{\sum_{i=1}^{N} L(f(\underline{x}_i), y_i)}{N}$$

average loss for classifier f over training set τ

For 0-1 loss,

$$Err_{train}(\tau) = \frac{\#\text{ data in } \tau \text{ labelled incorrectly by } f}{N}$$

It is easy to design f to have $Err_{train}(\tau) = 0$

Example $y \in \{-1, +1\}$, 0-1 loss

$$f(\tilde{\underline{x}}) = \begin{cases} \tilde{y} & \text{if } (\tilde{\underline{x}}, \tilde{y}) \in \tau \\ -1 & \text{otherwise} \end{cases}$$

OVER FITTING

But f may be terrible on data outside τ.

Prediction Error and Validation Error

$$Err_{pred} = E\left[L(f(\underline{X}), Y) \mid f \text{ is trained on } \tau\right]$$

↑ "average" over all possible $(\underline{x}, y)$

Estimate using another labelled data set independent of τ, Validation set $\mathcal{V} = \{(\tilde{\underline{x}}_1, \tilde{y}_1), \ldots, (\tilde{\underline{x}}_V, \tilde{y}_V)\}$

$$\hat{Err}_{pred}(\mathcal{V}) = \frac{\sum_{i=1}^{V} L(f(\tilde{\underline{x}}_i), \tilde{y}_i)}{V} = Err_{val}(\mathcal{V})$$

For 0-1 loss,

$$Err_{val}(\mathcal{V}) = \frac{\#\text{ data in } \mathcal{V} \text{ labelled incorrectly by } f}{V}$$

Displaying Error Performance

- Focus on 0-1 loss
- Err_{train} and Err_{val} measure overall error rate on training and validation datasets.
- More detailed measures of error may be needed for some applications (e.g. cancer detection)

Confusion Matrix

Focus on special case of binary classification

Classifier Output \ True Class	+1	-1
+1	TP: True Positives	FP: False Positives
-1	FN: False Negatives	TN: True Negatives
total	P	N

P = # data with +1 label

N = # data with -1 label

$N_{tot} = N + P$

True Positive Rate $TPR = \frac{TP}{P}$

False Positive Rate $FPR = \frac{FP}{N}$

True Negative Rate = specificity $= 1 - FPR$

$$\text{Precision} = \frac{TP}{TP + FP}$$

$$\text{Accuracy} = \frac{TP + TN}{N_{tot}} = 1 - Err$$

K-Nearest Neighbor classifier

Distance / Dissimilarity Measure

$\Delta(\underline{x}_1, \underline{x}_2)$ measures distance between feature vectors $\underline{x}_1$ and $\underline{x}_2$

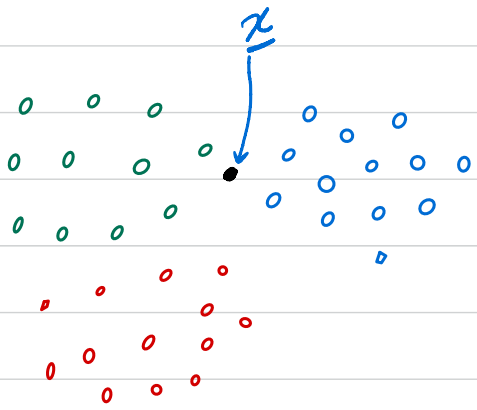
Examples: 1) $\Delta(\underline{x}_1, \underline{x}_2) = \|\underline{x}_1 - \underline{x}_2\|$

$= \sqrt{(\underline{x}_1 - \underline{x}_2)^T (\underline{x}_1 - \underline{x}_2)}$

2) $\Delta(x_1, x_2) = 1 - \cos(\angle(\underline{x}_1, \underline{x}_2))$

↑ angle

K-NN



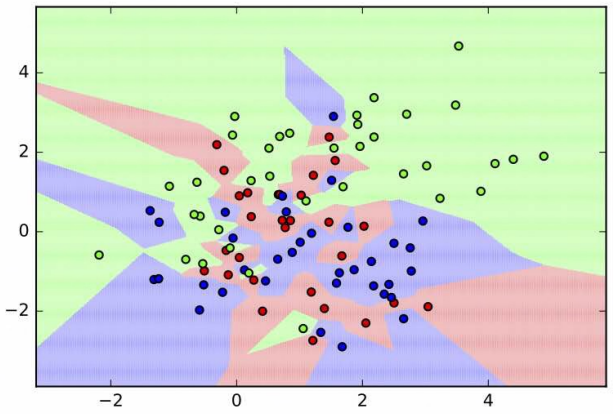
$k = 5$

$\underline{x}$ – green •

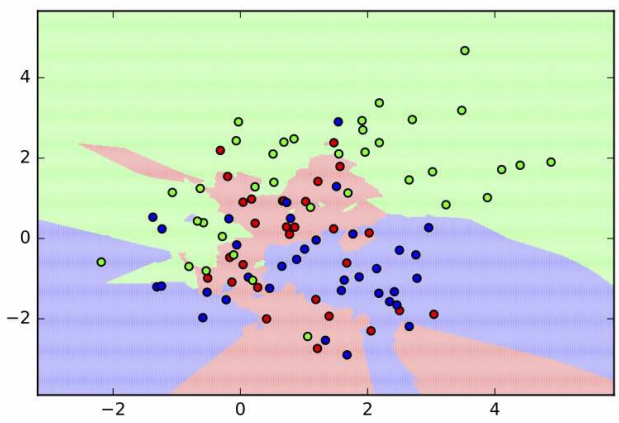
For each $\underline{x}$ pick k nearest neighbors using $\Delta(\cdot,\cdot)$ and choose label as most popular label.

For $k = 1$, algorithm is simply called NN classifier

NN



5-NN



Binary Bayes Classifier $\underline{x} \longrightarrow y \in \{+1, -1\}$

Suppose we know:

$$\pi_y \triangleq p(y) = P\{Y = y\}, \quad y = +1, -1$$

and $p(\underline{x} \mid y)$, $y = +1, -1$.

Then, binary classification problem is equivalent to binary hypothesis testing:

$H_{-1}: \underline{X} \sim p(\underline{x} \mid -1)$

$H_1: \underline{X} \sim p(\underline{x} \mid 1)$

priors π_{-1} and π_1

Joint distribution $p(\underline{x}, y) = p(\underline{x} \mid y)\, \pi_y$

For 0-1 loss,

$$\text{Err}_{\text{pred}} = P_e = \pi_{-1} \underbrace{P\{\text{say } 1 \mid -1 \text{ true}\}}_{P_{FA}} + \pi_1 \underbrace{P\{\text{say } -1 \mid 1 \text{ true}\}}_{P_M}$$

MAP Rule

$$\frac{p(\underline{x} \mid 1)}{p(\underline{x} \mid -1)} \underset{\text{say } -1}{\overset{\text{say } 1}{\gtrless}} \frac{\pi_{-1}}{\pi_1} \equiv \pi_1 p(\underline{x} \mid 1) \underset{\text{say } -1}{\overset{\text{say } 1}{\gtrless}} \pi_{-1} p(\underline{x} \mid -1)$$

$$\hat{y}_{MAP} = \arg\max_y \pi_y p(\underline{x} \mid y)$$

MAP rule is also called Bayes classifier

$$\hat{y}_{Bayes} = f_{Bayes}(\underline{x}) = \hat{y}_{MAP}$$

f_{Bayes} minimizes Err_{pred} — optimal, but...