MACQUARIE University

DATA SCIENCE

COMP2200/6200

12 – Second Half Summary (Week 7~11)

Xuyun Zhang (School of Computing)

Session 2 LEU Survey

MACQUARIE University

Provide feedback on teaching and course units

More are expected!

COMP2200	COMP	INT	31	495	6%
COMP6200	COMP	INT	36	413	9%

Image source: https://teche.mq.edu.au/2022/09/myth-busting-the-reporting-of-learner-experience-of-unit-surveys/

Week-8 Summary

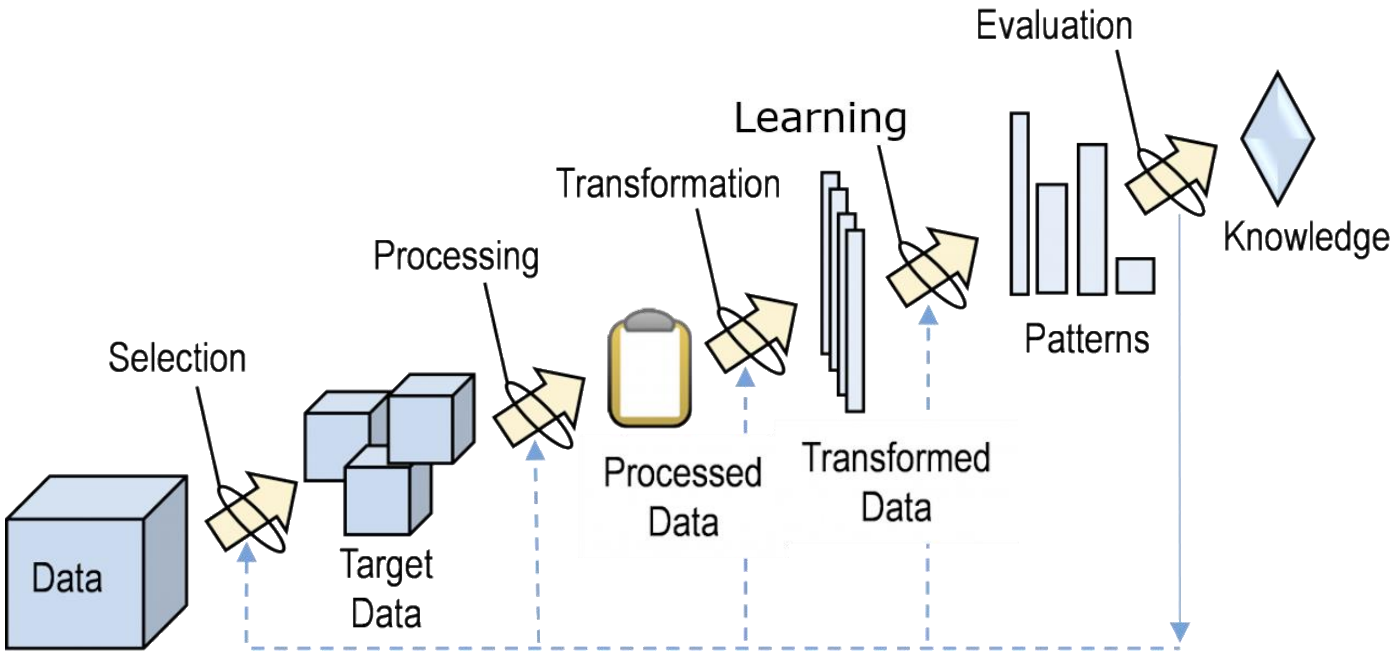
MACQUARIE University

- Learning and machine learning
- Data, model, and parameter spaces
- Supervised learning vs unsupervised learning
- K-NN classifier
- Model selection and model complexity
- Training/testing error
- k-fold cross validation
- Automatic hyperparameter tuning
- Performance Metrics

Data Mining and ML

MACQUARIE University

- Data Mining ≈ Database + Machine Learning
  - Database: data management and pre-processing
  - Machine learning: data analysis & knowledge discovery



What Is Machine Learning?

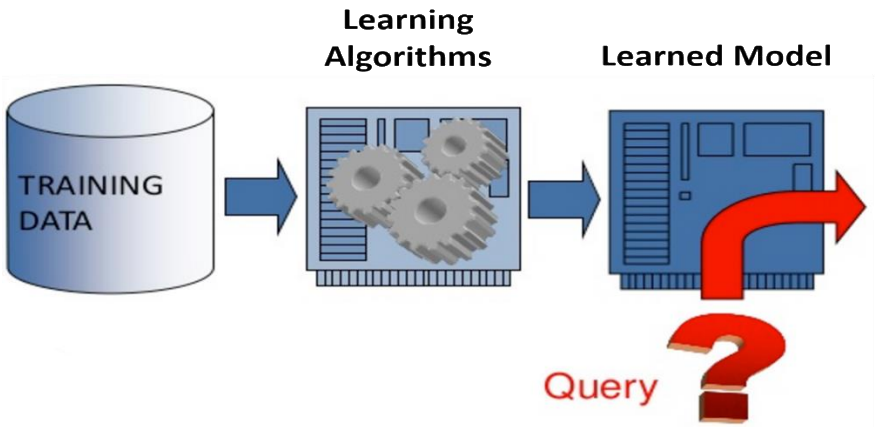
MACQUARIE University

- A subset of artificial intelligence in the field of computer science
  - Gives computers the ability to "learn" (i.e., progressively improve performance on a specific task) with data
  - Without being explicitly programmed
- Ingredients
  - **Model**: to represent the form of learning result
  - **Learning algorithm**: how to generate a model from data
  - **Data**: input to ML algorithms
- Aims
  - **to predict unseen data** or **to interpret existing data**

Two Stages of Machine Learning

MACQUARIE University

- Training stage
  - Training: to generate a model from observed data
- Testing stage
  - Testing: to use the learned model to predict unseen data

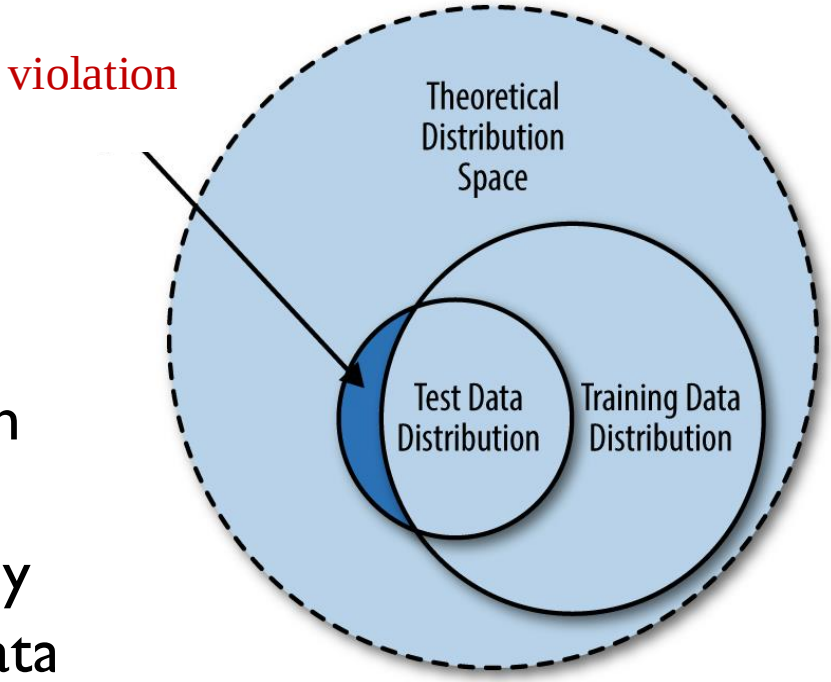


E.g., does a patient with coughing, running nose and fever suffer from COVID-19 or flu?

Why ML Works & Why Not

MACQUARIE University

- Assumption: the distribution of training examples is identical to the distribution of test examples (including future unseen examples)
  - In practice, this is often violated to certain degree
  - Strong violations will clearly result in poor performance
  - To achieve good accuracy on the test data, training examples must be sufficiently representative of the test data



Supervised vs Unsupervised

MACQUARIE University

- Label/target of data
  - The interesting attribute(s) for prediction
  - Further, $d_i \equiv \langle x_i, y_i \rangle$
  - **Supervised learning models**
    - **Regression**: y is continuous
    - **Classification**: y is discrete (binary or multi-class)
- No explicit label/target information
  - Then, $d_i \equiv \langle x_i, \cdot \rangle$
  - **Unsupervised learning models**
    - ***Clustering***
- **Semi-supervised learning** (labeling is often costly)

Model

MACQUARIE University

- **Model**: a map from input space to output space
  - i.e., $f : \mathcal{X} \to \mathcal{Y}$
  - An infinite number of such maps
  - Input space: space spanned by feature attributes
  - Output space: space spanned by label/target attributes
- **Hypothesis space** H: space of all possible maps
  - Functional space: $\mathcal{H} \equiv \{f \mid \mathcal{X} \to \mathcal{Y}\}$
- **Ground truth**: underlying true mechanisms of generating the observed data
  - But never known in reality
  - The purpose of learning: to approximate the ground truth

Parameter Space

MACQUARIE University

- What do we really learn from data for a model?
  - Hypothesis space is usually pre-specified in terms of problem domains
  - Different models are determined by parameters
  - Let θ denote the parameter vector, we have

$$\mathcal{H} \equiv \{f \mid y = f_\theta(x)\}$$

- Parameter space
  - Can be real value spaces, e.g., $\mathbb{R}^n$
  - Structure of a model (more implicit), e.g., tree or graph structures, as well as partitions of the input space

Model Selection?

MACQUARIE University

- **Model Selection**
  - Multiple models generated by different algorithm parameter configurations
  - Multiple models generated by different learning algorithms
- **Model complexity**
  - Very generally, it refers to the number of degrees of freedom in a learned model
  - Often measured as the number of adjustable weights or parameters in the architecture, e.g., weights in regression
  - High complexity → stronger capability of capturing information from training data

Training/Testing Errors

MACQUARIE University

- **Training error** (or empirical error) of a trained model $\hat{f}$ on a training data set of size N

$$E_{emp}(\hat{f}) = \frac{1}{N}\sum_{i=1}^{N} L(y_i, \hat{f}(x_i))$$

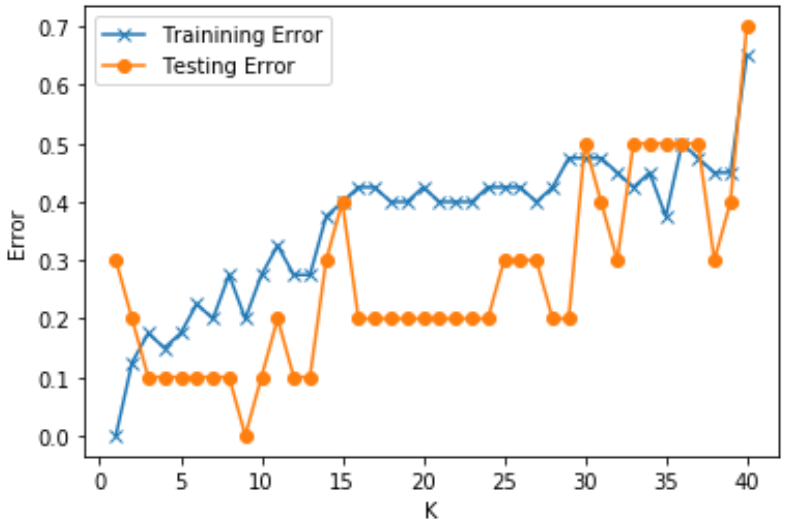
  - Note that the loss function $L(\cdot,\cdot)$ requires instantiation for a specific model, e.g., the squared error in linear regression

- **Testing error** on a test data set with size N'

$$E_{test}(\hat{f}) = \frac{1}{N'}\sum_{i=1}^{N'} L(y_i, \hat{f}(x_i))$$

  - Indicating the generalisation capability of a learned model

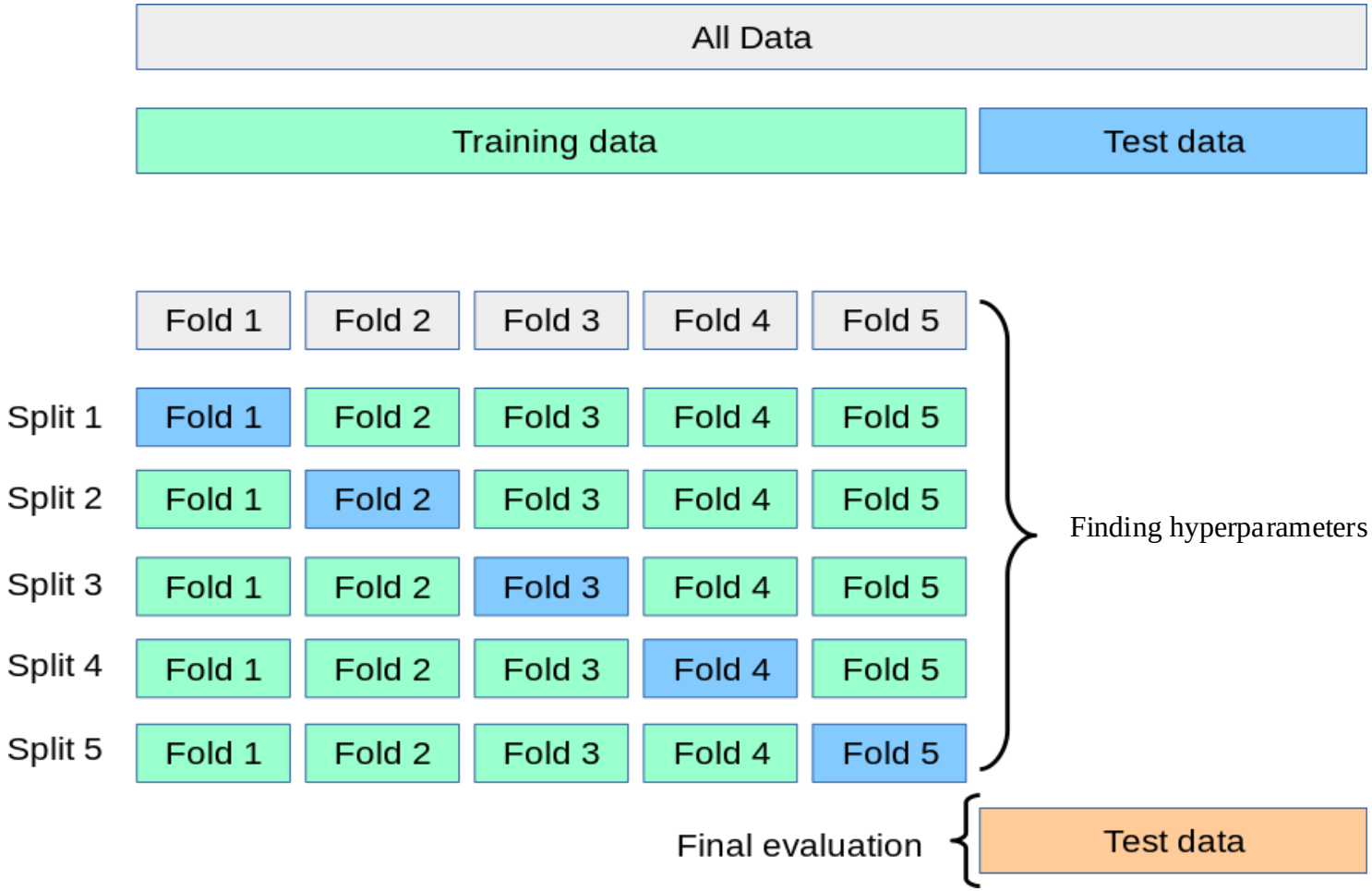
Overfitting vs Underfitting



- Overfitting: a model is too strong and captures the very details of training samples, lacking generalization capability
  - Data is noisy, and the model is fitted to noise
  - Fighting against overfitting is important in machine learning!
- Underfitting: increasing model complexity can help

k-fold Cross Validation

MACQUARIE University

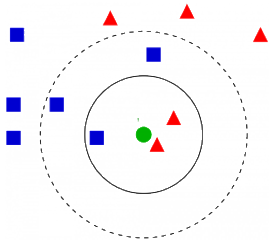


MACQUARIE University

DATA SCIENCE

COMP2200/6200

08 – K-Nearest Neighbors Classifier



Xuyun Zhang (School of Computing)

Nearest Neighbors Classification

MACQUARIE University

- Idea: instance-based learning
  - Similar examples have similar labels
  - Classify new examples like similar training examples
- Algorithm
  - Given some new example $\boldsymbol{x}$ for which we need to predict its class y
  - Find the most similar training examples
  - Classify $\boldsymbol{x}$ "like" these most similar examples
- Questions:
  - How to determine similarity?
  - How many similar training examples to consider?

KNN Limitations

MACQUARIE University

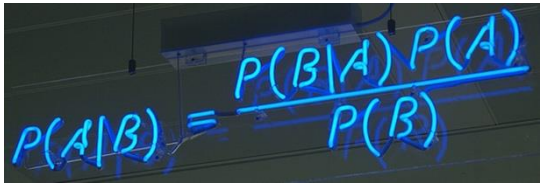
- Each prediction takes $O(n)$ computational complexity
  - Use fancy data structures such as KD-trees to accelerate the search of nearest neighbours
  - Or use locality-sensitive hashing to approximate nearest neighbours with constant computational complexity
- Prediction performance degrades when number of attributes grows
  - Curse of dimensionality: when the number of attributes is big, similarity/distance measures become less reliable
  - Remedy
    - Remove irrelevant attributes in pre-processing
    - Weight attributes differently

MACQUARIE University

DATA SCIENCE

COMP2200/6200

09 – Naïve Bayes Classifier



Xuyun Zhang (School of Computing)

How to Get Probability?

MACQUARIE University

- If we assume that each outcome occurs with an equal chance, the probability is $\frac{1}{|\Omega|}$ (theoretical probability)
  - E.g., fair coin flipping, $\Pr\{X = H\} = \Pr\{X = T\} = 1/2$
  - E.g., fair die tossing, $\Pr\{X = 1\} = \cdots = \Pr\{X = 6\} = 1/6$
- Estimation from relative frequency
  - Repeat random experiments N times
  - Count the occurrence/frequency of an outcome x, N_x
  - Relative frequency $\frac{N_x}{N}$
  - Estimate $\Pr\{X = x\} = \lim_{N \to +\infty} \frac{N_x}{N}$ (law of large numbers)
  - In practice, make N large enough to have a good estimation

Probabilistic Modelling

MACQUARIE University

- How can we model classification with probability?
- Each attribute is a random variable
  - Target is a discrete random variable with distribution $P(\mathcal{C})$
  - Features are a random vector with joint distribution $P(X)$
  - Whole data can be captured by joint distribution $P(X, \mathcal{C})$
- Then, classification problem is to estimate $P(\mathcal{C}|X = \boldsymbol{x})$
  - $\boldsymbol{x}$ is the feature vector of a testing instance
  - Distribution $P(\mathcal{C}|\boldsymbol{x})$ can tell the probability of each class label
  - The label with the highest probability is the predicted label

$$\text{Label}(x) \leftarrow \arg\max_{\mathcal{C}_k} \{p(\mathcal{C}_k|\boldsymbol{x})\}$$

    - This can minimize classification error

Probabilistic Modelling (Cont'd)

MACQUARIE University

- The key is to estimate conditional distribution $P(\mathcal{C}|X)$
  - Question: is $\mathcal{C}$ and X independent ?
- In terms of product rule

$$P(\mathcal{C}|X) = \frac{P(\mathcal{C},\ X)}{P(X)}$$

- Option 1: estimate joint distribution $P(\mathcal{C},\ X)$ directly
  - Then, $P(X)$ can be estimate by sum rule

$$P(X) = \sum_{\mathcal{C}} P(\mathcal{C}, X)$$

  - This is Bayes optimal classifier

Naïve Bayes Classifier

MACQUARIE University

- Challenge of estimating $P(X|C)$: still need to model the joint probability of all features of $\boldsymbol{x} = \langle x_1, x_2, \ldots, x_M \rangle$
  - Many parameters to estimate (much more than # of features)
- Naïve Bayes assumption: all input features are conditionally independent of each other
  - Strong assumption (hard to achieve in reality)

$$
\begin{aligned}
p(\boldsymbol{x}|\mathcal{C}_k) &= p(x_1, \cdots, x_M|\mathcal{C}_k) \\
&= p(x_1|x_2, \cdots, x_M, \mathcal{C}_k)p(x_2, \cdots, x_M|\mathcal{C}_k) \\
&= p(x_1|\mathcal{C}_k)p(x_2, \cdots, x_M|\mathcal{C}_k) \\
&= p(x_1|\mathcal{C}_k) \cdots p(x_M|\mathcal{C}_k) = \prod_{i=1}^{M} p(x_i|\mathcal{C}_k)
\end{aligned}
$$

Product of individual probabilities

Naïve Bayes Classifier (Cont'd)

MACQUARIE University

- Now comes the decision rule

Classification is easy, just have probabilities multiplied

$$Label(\boldsymbol{x}) \leftarrow \arg\max_{C_k} P(C_k) \prod_{i=1}^{M} p(x_i|C_k)$$

- Scalable (linear # of parameters w.r.t. M features)
- In practice, we just need to simply compute the relative frequencies from training data to estimate the probabilities $P(C_k)$ and $p(x_i|C_k)$
- Performance is comparably good even though the conditionally-independent assumption is very strong

Example: Training/Learning

MACQUARIE University

PlayTennis: training examples

Day	Outlook	Temperature	Humidity	Wind	PlayTennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

P(Play=*No*) = 5/14

P(Play=*Yes*) = 9/14

Outlook	Play=*Yes*	Play=*No*
Sunny	2/9	3/5
Overcast	4/9	0/5
Rain	3/9	2/5

Temp.	Play=*Yes*	Play=*No*
Hot	2/9	2/5
Mild	4/9	2/5
Cool	3/9	1/5

Wind	Play=*Yes*	Play=*No*
Strong	3/9	3/5
Weak	6/9	2/5

Humidity	Play=*Yes*	Play=No
High	3/9	4/5
Normal	6/9	1/5

NBC Pros and Cons

MACQUARIE University

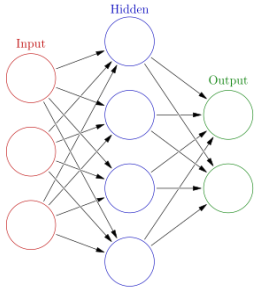
- Pros
  - Very simple, and easy to implement.
  - Work well in practice even if NB assumption doesn't hold.
  - Highly scalable and fast, as it scales linearly with the number of features and data instances.
  - Can be used for both binary and multi-class classification.
  - Can make probabilistic predictions.
  - Can handle both continuous and discrete attributes.
  - Insensitive to irrelevant features.
- Cons
  - Strong assumption on NB conditional independence : any two features are independent given the output class.

MACQUARIE University

DATA SCIENCE

COMP2200/6200

10 – Artificial Neural Networks

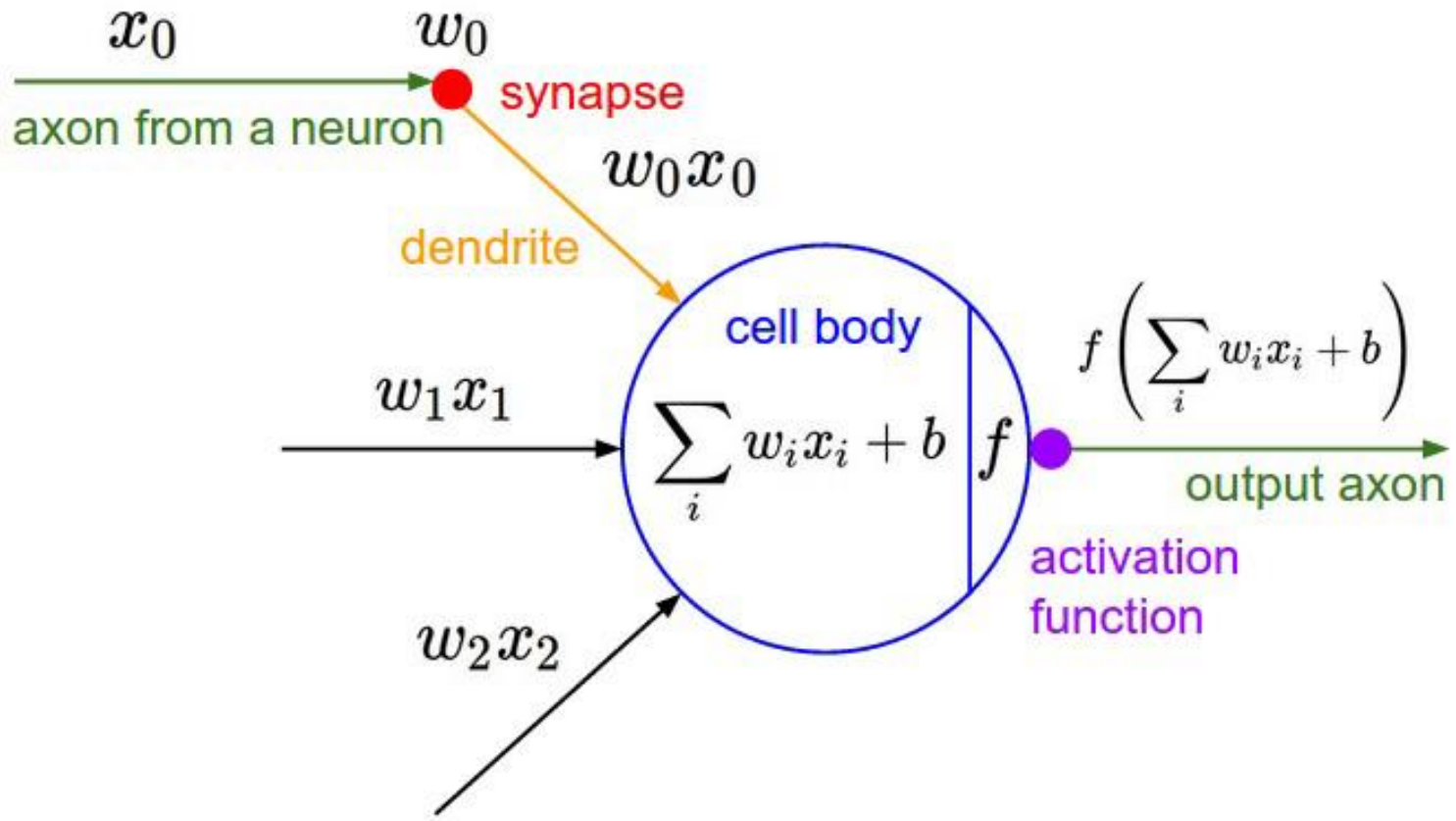


Xuyun Zhang (School of Computing)

Artificial Neuron

MACQUARIE University

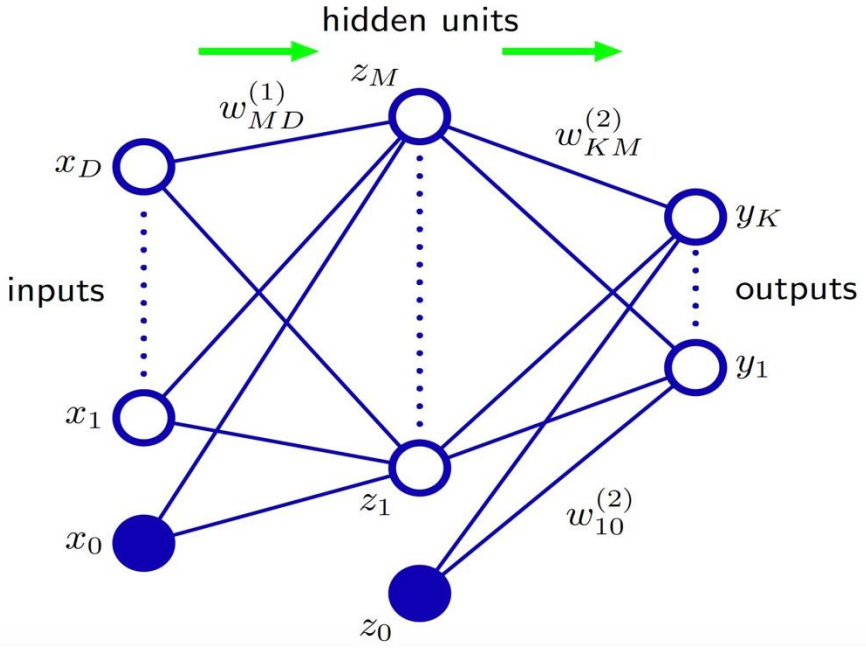
- Computational model
  - Activation function on weighted sum of inputs



Neural Network Layers

MACQUARIE University

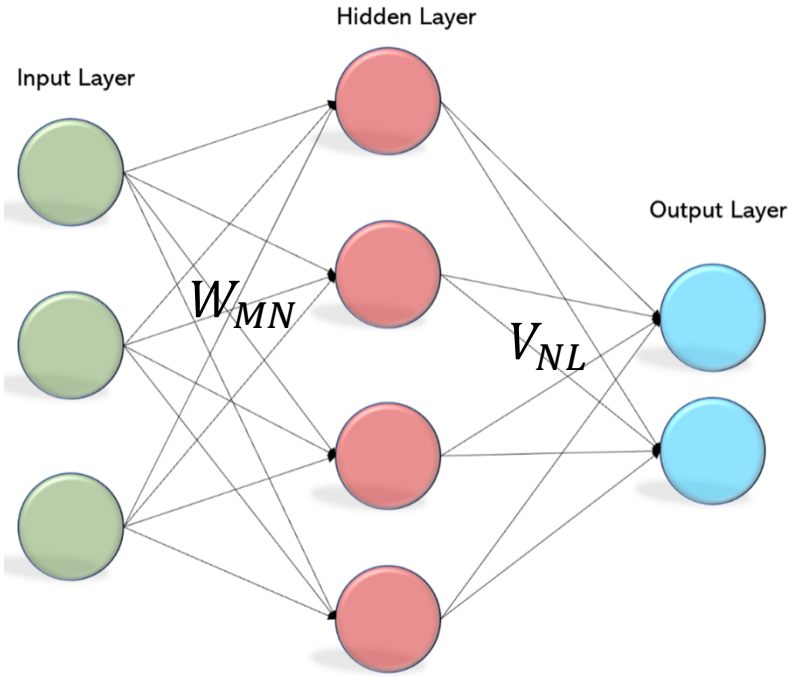
- Input layer
  - Determined by input features, usually one unit for one feature
- Output layer
  - Determined by output results
  - E.g., one unit for regression, multiple units for classification
- Hidden layer(s)
  - Determined by users
  - # of layers
  - # of units in each layer
  - How to connect units
  - Which activation function to use



Multi-Layer Perceptron

MACQUARIE University

- One or more hidden layers
- Here we consider one hidden layer
- Consider following setting
  - # Input feature: M
  - # Output dimension: L
  - # Hidden units: N
  - Weights: W_{MN} and V_{NL}
  - Activation function h
  - Output function g
  - Note: bias terms included for denotation convenience



- Model: $\boldsymbol{y} = \boldsymbol{f}(\boldsymbol{x}|W, V) = \boldsymbol{g}(\boldsymbol{h}V_{NL}) = \boldsymbol{g}(\boldsymbol{h}(\boldsymbol{x}W_{MN})V_{NL})$

Model Training

MACQUARIE University

- What to learn?
  - Model parameters: W, V
  - # of model parameters: $M * N + N * L$
- Network topology is a hyperparameter
  - Not a model parameter
  - Determine the number of model parameters
  - Control model complexity
  - Note that activation functions also affect model complexity
- How to learn?
  - Minimize cost function w.r.t. a data set to obtain W, V
  - An optimization problem

Error Back Propagation

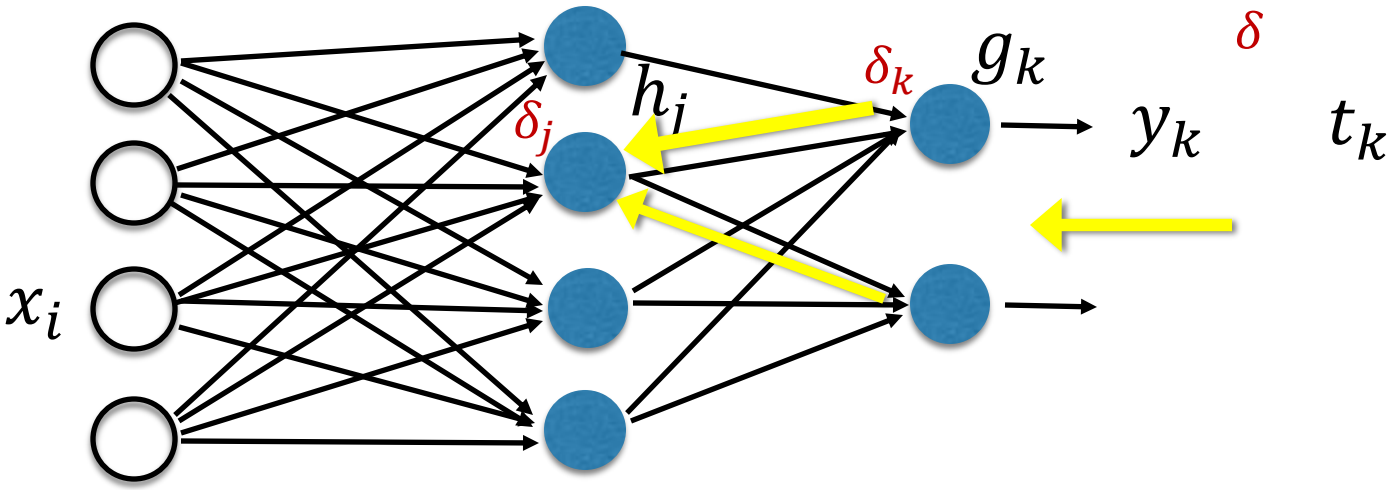
MACQUARIE University

- Partial derivatives (can be used for weight update)
  - $\frac{\partial E_n(W,V)}{\partial v_{jk}} = \sum_{r=1}^{n} \left(g_k - t_k^{(r)}\right) g_k' h_j$
  - $\frac{\partial E_n(W,V)}{\partial w_{ij}} = \sum_{r=1}^{n} (\sum_{k=1}^{L} \left(g_k - t_k^{(r)}\right) g_k' v_{jk}) h_j' x_i^{(r)}$
- Observations (consider a single data instance update)
  - Let error signal $(g_k - t_k) \equiv \delta$
  - For the output layer: $\frac{\partial E_n(W,V)}{\partial v_{jk}} = \delta_k h_j$, $\delta_k \equiv \delta g_k'$
  - For hidden layers: $\frac{\partial E_n(W,V)}{\partial w_{ij}} = \delta_j x_i$, $\delta_j \equiv (\sum_{k=1}^{L} \delta_k v_{jk}) h_j'$
  - Error signal δ_j is calculated from δ_k, i.e., **back propagation**!

Error Back Propagation (Cont'd)

MACQUARIE University

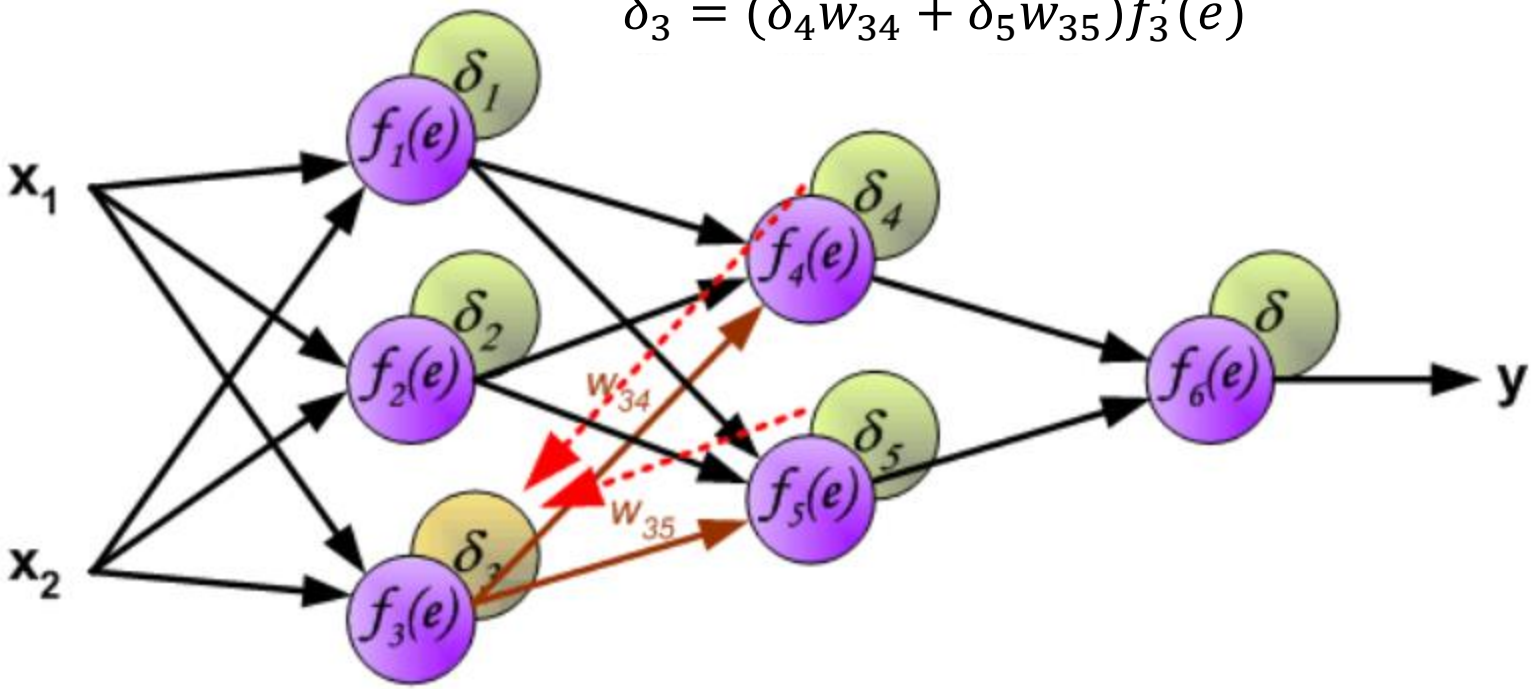
- Back propagation of error signals
  - $\delta = (g_k - t_k)$. Note: this error will be different for other cost functions, e.g., cross-entropy.
  - $\delta_k = \delta g'_k$
  - $\delta_j = (\sum_{k=1}^{L} \delta_k v_{jk}) h'_j$



BP Example (Cont'd)

MACQUARIE University

$$\delta_3 = (\delta_4 w_{34} + \delta_5 w_{35}) f_3'(e)$$

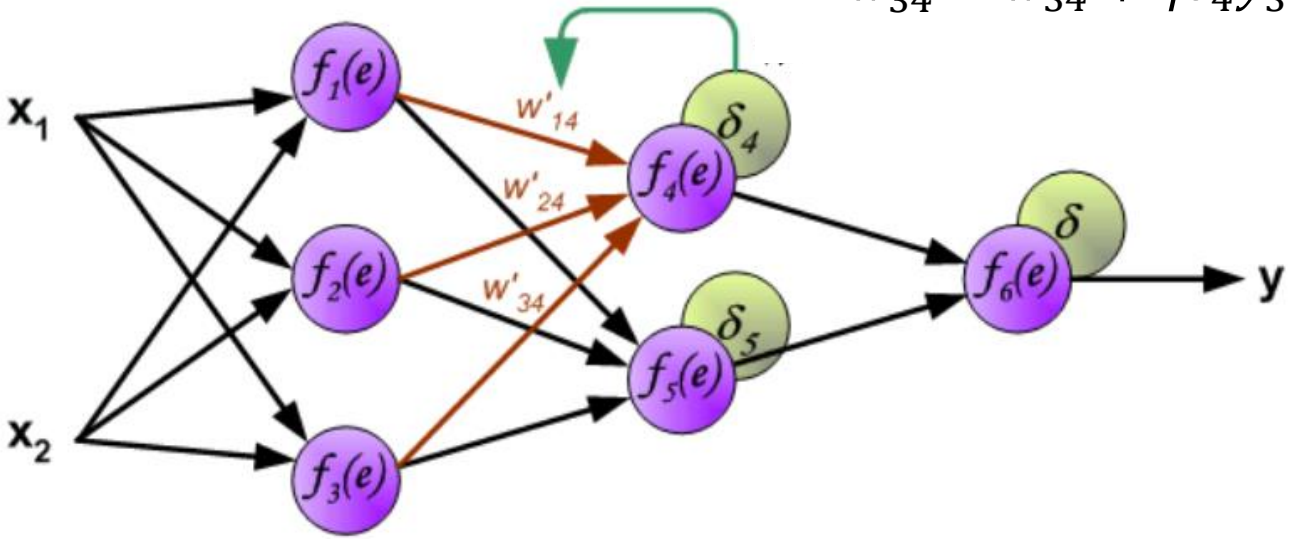


BP Example

$$w'_{14} = w_{14} + \eta\delta_4 y_1$$

$$w'_{24} = w_{24} + \eta\delta_4 y_2$$

$$w'_{34} = w_{34} + \eta\delta_4 y_3$$



ANN Pros and Cons

MACQUARIE University

- Pros
  - The model is powerful with high accuracy and can be applied to complex non-linear problems.
  - Can work with large-scale datasets (big data).
  - Prediction is fast.
  - No need for feature engineering in deep learning models.
  - Now many deep learning frameworks, e.g., PyTorch.
- Cons
  - Model training is computation intensive, e.g., GPUs are often required for deep learning models.
  - Data intensive, otherwise, overfitting is likely to occur
  - Poor interpretability, i.e., difficult in explaining the prediction

MACQUARIE University

DATA SCIENCE

COMP2200/6200

11 – Decision Tree

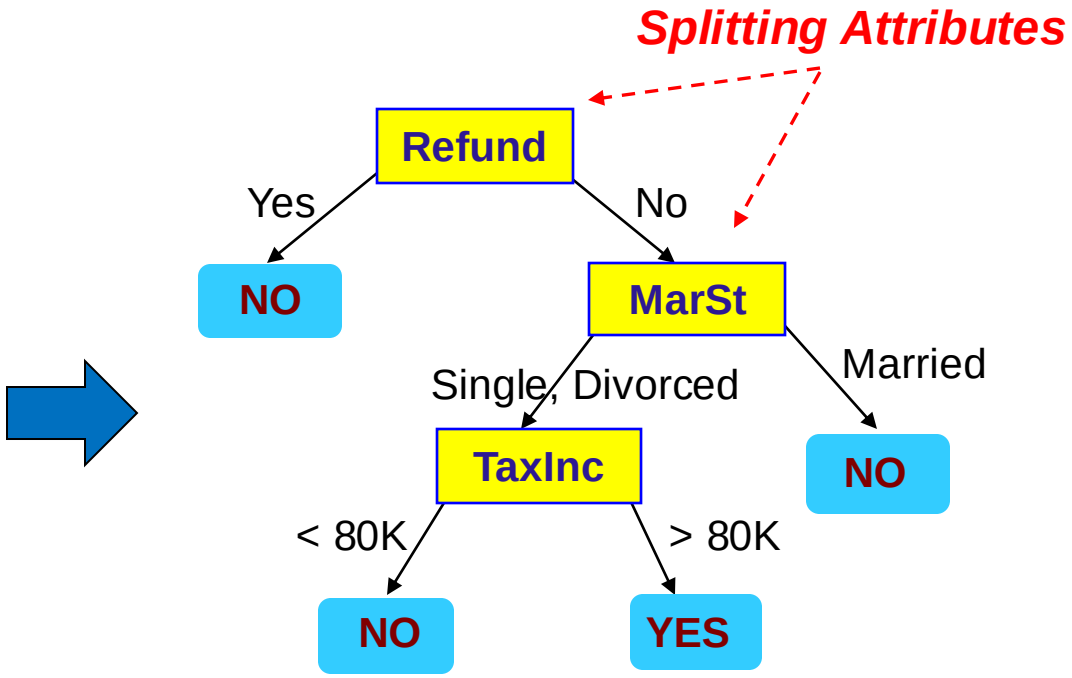
Xuyun Zhang (School of Computing)

Training

- Training: construct a (decision) tree structure
  - Decision nodes: tracking partitioning (tree structure)
  - Terminal nodes: representing decision regions

Tid	Refund	Marital Status	Taxable Income	Cheat
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

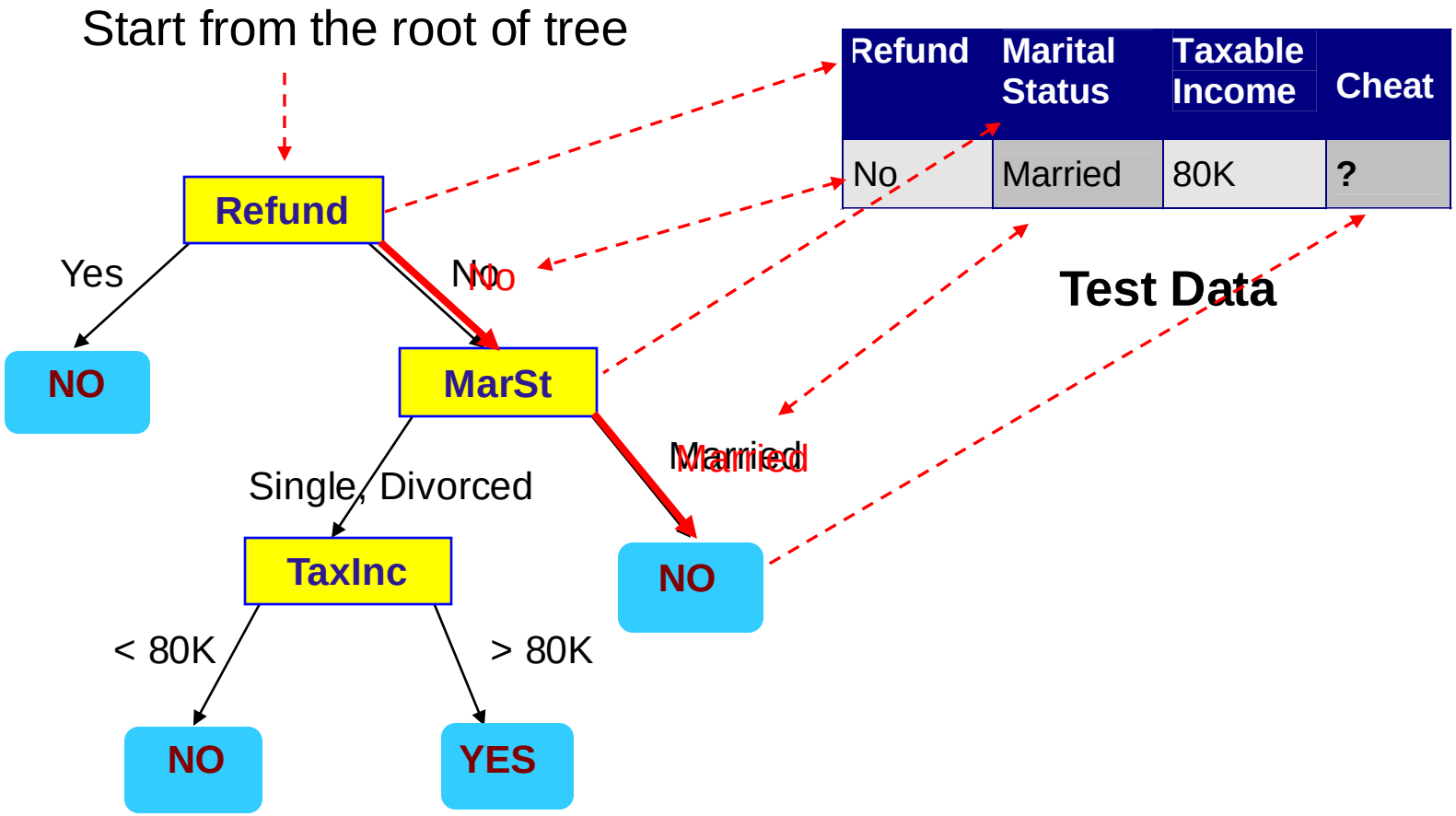
Training Data



Model: Decision Tree

Testing

- Predicting: traverse the built decision tree for a region



How to Partition Data ?

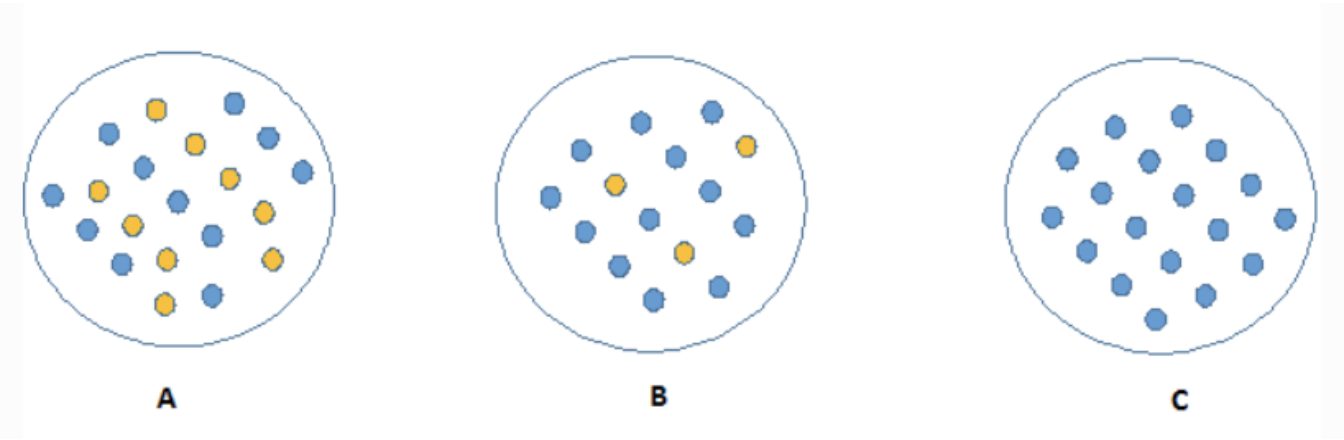
MACQUARIE University

- Two fundamental questions!
- Which axis (dimension/feature) should be chosen?
  - To minimise classification error
  - Computationally infeasible to try all possible tree structures
    - E.g., for 10 binary features, search space size is 2^{10}
  - Greedy optimisation: use a heuristics
    - E.g., information gain
- What values of the chosen attribute for splitting?
  - Depending on data types: categorical or continuous
  - Continuous features (or dimension) need discretisation
  - Binary or multiway tree structure

Splitting Heuristics

MACQUARIE University

- Which partition are more expected after splitting?



- Answer: C. Why?
  - No/few splits are required
  - Voting for class label is easier (similar to KNN classifier)
  - Impurity refers to this quality
    - Entropy (information), Gini impurity, etc.

Information Gain

MACQUARIE University

- Assume a data set D, the distribution of label $\boldsymbol{t}_D$ is

Class	C_1	$\cdots$	C_K	Total
Count	c_{p1}	$\cdots$	c_{pK}	N_D

  - Information (entropy) of $\boldsymbol{t}_D$: $Info(\boldsymbol{t}_D) = -\sum_{i=1}^{K} p_i \log_2(p_i)$
  - Probability $p_i = c_{pi}/N_D$
- Partition D by feature ϕ_j, producing $\{D_1, \ldots, D_v\}$
  - $\boldsymbol{t}_D$'s information after partition:

$$Info_{\phi_j}(\boldsymbol{t}_D) = \sum_{i=1}^{V} \frac{N_{D_i}}{N_D} Info(D_i)$$

- Information gain of this partition

$$Gain_D(\phi_j) = Info(\boldsymbol{t}_D) - Info_{\phi_j}(\boldsymbol{t}_D)$$

Gain Ratio

MACQUARIE University

- Info gain issue: prefer attributes with more values
  - Extreme case: prefer unique identifiers e.g., instance IDs
- Solution
  - Normalisation on information gain for each feature
- Split information of a feature ϕ_j (w.r.t. original dataset)

$$SplitInfo_{\mathcal{D}}(\phi_j) = -\sum_{i=1}^{V} \frac{N_{\mathcal{D}_i}}{N_{\mathcal{D}}} \log_2\left(\frac{N_{\mathcal{D}_i}}{N_{\mathcal{D}}}\right)$$

- Gain Ratio: normalised information gain

$$GainRatio_D(\phi_j) = Gain_D(\phi_j) / SplitInfo_{\mathcal{D}}(\phi_j)$$

Gini impurity

MACQUARIE University

- Gini impurity
  - Probability of two samples having different labels when being randomly chosen from a dataset

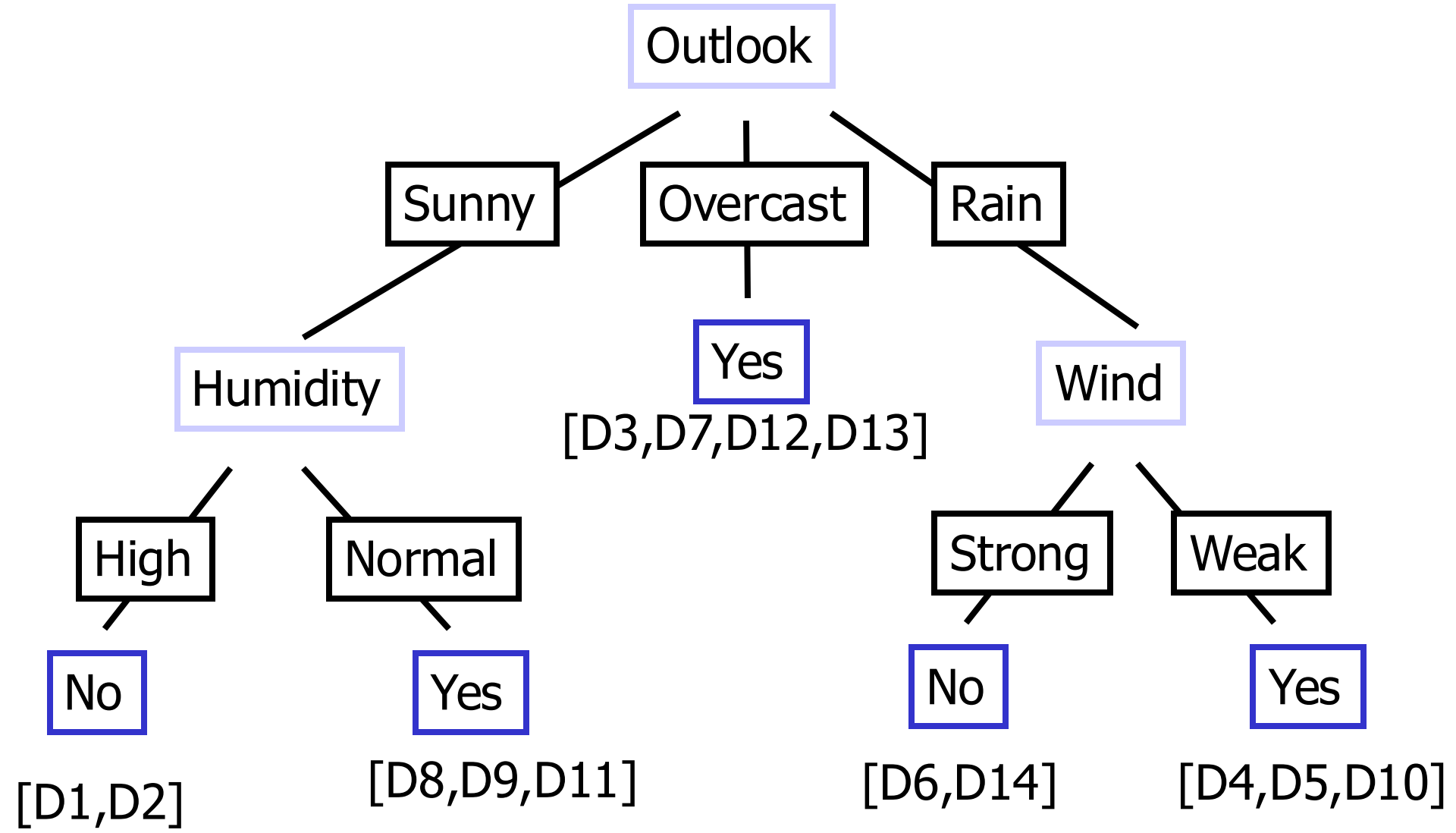
$$Gini(D) = 1 - \sum_{i=1}^{K} p_i^2 = \sum_{i=1}^{K} p_i(1 - p_i)$$

  - Partition D by feature ϕ_j

$$Gini_D(\phi_j) = \sum_{i=1}^{V} \frac{N_{D_i}}{N_D} Gini(D_i)$$

- The lower the better for splitting
  - Zero Gini impurity implies perfect classification
- Selection of splitting heuristic is data dependent

Example: Algorithm (Cont'd)



Decision Tree Pros and Cons

MACQUARIE University

- Pros
  - Simplicity & interpretability: easy to understand and interpret
  - Can work with both numerical and categorical data
  - Capture non-linear relationships
  - Provide insight into feature importance
  - Fast to train and predict
- Cons
  - Overfitting: can easily become too complex
  - The splitting decisions are made based on local optimality, which may not lead to the global best tree structure
  - Lack of Smoothness: piecewise constant predictions, which can lead to abrupt changes in predicted values.

Q & A

MACQUARIE University