Comp 90049 Intro to ML

Pei-Yun Sun (Stats, Data Science, CS)

Email: pssun@unimelb.edu.au

Concept: - what we're trying to predict / understand
- Output of the system
- label / classes (supervised learning)

Instance: single exemplar from data (consist of attribute values)

Attribute: single measurement of some aspect of an instance (features)

(i) Skin cancer screening test

Concept: Cancer / Not cancer (binary)

Instance: patient

Attribute: result of blood test, images from skin, reports, observed syndromes, ...

S: (binary) Classification

Generalisation:

training data often have biases, e.g. skin cancer risk often increase with ages

⇒ correlated in training set

- Good: the model could learn age $\xrightarrow{\text{predicts}}$ skin cancer
- Bad: if too dependent on age ⇒ may not work well on young patients if there were very few instances of younger patients

(ii) Building a system that guesses what the weather (temperature, precipitation, etc.) will be like tomorrow

Concept : Weather : quantity e.g. temperature , precipitation , humidity , UV index ...

Instance : A day

Attribute : data from previous days

S : Regression (numeric)

↑ e.g. temperature

Generalisation : might work better in some cities than others

better if included geographic info

new features → weather patterns interact with

(iii) Predicting products that a customer would be interested in buying, based on other purchases that customer has previously made

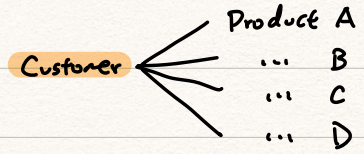
(a) Customer - Product pairs

Concept : Interested / Not interested

Instance : Pair

S : classification (I/NI)

(b)



Concept : Products

Instance : Customer

Attributes : name , age , shopping log , gender , ...

US : clustering / association rule mining (attr of prod → buy / not)

(groups of customers)

Generalisation: - Customer model in one country might not generalise to other countries

- if it learns everyday shopping patters

⇒ may not work for outlier situations e.g. holiday purchasing

Supervised: - instances labelled with classes (training data)

- Classify / predict instances in test data (no labels)

Un-S: Not based on labelled traiig data (ignore)

↑ find hidden patterns / groups

5. What kinds of assumptions might a machine learning model make when tackling these problems?

① Concept is actually related to the attributes (obvious!)

- We only include attributes we think are likely to predict the concept
- e.g. you won't use attributes like "patient's favorite song" as an attribute for skin cancer detection

BUT song —(good predictor)→ age —(risk factor)→ skin cancer

↑ might be a good predictor

② Each model makes assumptions about the ways the attributes can relate to concepts

e.g.

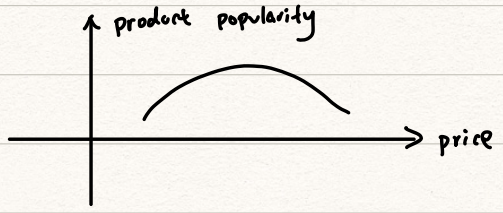
(i)
- treat all attributes as independent predictors ✓
- Allow predictors to interact ⇒ could lead to an overly complex model if there are many attributes to start with

(ii) Numerical attributes:

- Generally expect linear / monotonic relationships between attributes & concepts

good simplifying assumption but limits what the model can learn

e.g. Price - Product (U-shape)



6. What is **discretisation**, and where might it be used? Discretise attribute C of the following dataset according to the given methods (breaking ties where necessary).

(i) Equal width

(ii) Equal frequency

(iii) k-means

ID	A (°C)	B (mm)	C (hPa)	CLASS
1	22.5	4.6	1021.2	AUT
2	16.7	21.6	1027.0	AUT
3	29.6	0.0	1012.5	SUM
4	33.0	0.0	1010.4	SUM
5	13.2	16.4	1019.5	SPR
6	14.9	8.6	1016.4	SPR
7	18.3	7.8	995.4	WIN
8	16.0	5.6	1012.8	WIN

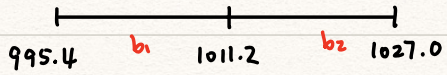
(i) Equal width

Divide the "range" of possible values seen in training set into equally-sized subdivisions (regardless of # instances in each division)

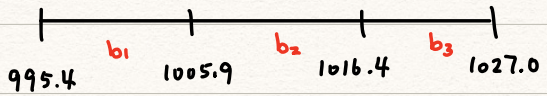
Example:

Attribute C: min = 995.4 max = 1027.0 range = 31.6

2 buckets: bucket width = $\frac{31.6}{2} = 15.8$



3 buckets: bucket width = $\frac{31.6}{3} \approx 10.5$



② Equal frequency

Divide the "range" of possible values seen in training set, s.t. (roughly) the same number of instances appear in each bucket

Example:

Attribute C:

Sort (Asc):

995.4 (7) < 1010.4 (4) < 1012.5 (3) < 1012.8 (8) < 1016.4 (6) < 1019.5 (5) < 1021.2 (1) < 1027.0 (2)

2 buckets:

b1: 7, 4, 3, 8

b2: 6, 5, 1, 2

E.g. $b_1 \rightarrow [995.4, \frac{1012.8+1016.4}{2})$

$b_2 \rightarrow [\frac{1012.8+1016.4}{2}, 1027.0]$

③ K-means

A clustering approaching. But can work well in this context.

To get k buckets:

1. Randomly choose k points to act as seeds.
2. Iteratively: Assign each instance to the nearest bucket
   & Update "centroid" of the bucket with mean of the values

Example:

Attribute C:

2 random seeds: 1012.5 (3) → Bucket A

1010.4 (4) → Bucket B

Assign: 1021.2 (1) → A

(To the nearest) 1027.0 (2) → A

1012.5 (3) → A

...

1012.8 (8) → A

⇒ A: 1, 2, 3, 5, 6, 8

B: 4, 7

Update: $C_A = \frac{1}{6}(1021.2 + \cdots + 1012.8) = 1018.2$

$C_B = \frac{1}{2}(1010.4 + 995.4) = 1002.9$

Repeat ... (Assign & Update until no changes to the buckets)

Final: A: 1, 2, 3, 5, 6, 8 (b_1)

B: 4, 7 (b_2)