Machine Learning
W4 Tutorial

COMP30027 | Sandy Luo

Overview

Data conversion

Discrete, continuous

Naive Bayes

Theory, calculation

Naive Bayes

code!

Consider the following dataset:

ID	Color	Weight (g)
1	Red	1021.2
2	Red	1027.0
3	Red	1012.5
4	Blue	1010.4
5	Blue	1019.5
6	Gold	1016.4
7	Gold	995.4
8	Gold	1012.8

instances / data points (rows)

attributes / features (columns)

attributes? instances?

Q1:

Which attribute is discrete and which is continuous?

Consider the following dataset:

ID	Color	Weight (g)
1	Red	1021.2
2	Red	1027.0
3	Red	1012.5
4	Blue	1010.4
5	Blue	1019.5
6	Gold	1016.4
7	Gold	995.4
8	Gold	1012.8

Q1:

Which attribute is discrete
and which is continuous?

Value set: {"Red", "Blue", "Gold"}

3 possible values
→ limited / finite
→ discrete

Value set: {1021.2, 1027.0, ...}

Many possible values
→ unlimited / infinite
→ continuous

Consider the following dataset:

ID	Color	Weight (g)
1	Red	1021.2
2	Red	1027.0
3	Red	1012.5
4	Blue	1010.4
5	Blue	1019.5
6	Gold	1016.4
7	Gold	995.4
8	Gold	1012.8

Discretisation Methods

1. **Equal-width**
   - Find min. & max. values
   - Partition into *n* bins of width *(max - min) / n*

2. **Equal-frequency**
   - Sort values
   - Split sorted values into n bins with equal numbers of items

Discretisation Methods

3. **Clustering (e.g. k-means)**
   - Randomly initialise *n* centre points for *n* clusters
   - Iteratively assign each data point to the nearest centroid
   - Update centroids as the mean of assigned points until convergence

4. **Supervised classification**
   - Use class labels to determine bin boundaries
   - Group values into class-contiguous intervals

Q2:

Discretise the continuous attribute into 2 bins using the (unsupervised) n = 2 methods of **equal width**, **equal frequency**, and **k-means** (break ties where necessary).

Consider the following dataset:

ID	Color	Weight (g)
1	Red	1021.2
2	Red	1027.0
3	Red	1012.5
4	Blue	1010.4
5	Blue	1019.5
6	Gold	1016.4
7	Gold	995.4
8	Gold	1012.8

Q2(a): n = 2

Equal-width:

- Find min. & max. values
- Partition into *n* bins of width *(max - min) / n*

- min = 995.4
- max = 1027.0
- width = (1027.0 - 995.4) / 2 = 15.8
- bins: [min, min+15.8), [min+15.8, max]
  - = [995.4, 1011.2), [1011.2, 1027.0]

Consider the following dataset:

ID	Color	Weight (g)
1	Red	1021.2
2	Red	1027.0
3	Red	1012.5
4	Blue	1010.4
5	Blue	1019.5
6	Gold	1016.4
7	Gold	995.4
8	Gold	1012.8

Q2(b): n = 2

Consider the following dataset:

ID	Color	Weight (g)
1	Red	1021.2
2	Red	1027.0
3	Red	1012.5
4	Blue	1010.4
5	Blue	1019.5
6	Gold	1016.4
7	Gold	995.4
8	Gold	1012.8

Equal-frequency:

- Sort values
- Split sorted values into *n* bins with equal numbers of items

- bin size = # instances / n
  = 8 / 2 = 4

- Sorted order (ascending):
  - ID: 7, 4, 3, 8, 6, 5, 1, 2
- 4 values in each bin, hence:
  - ID: [7, 4, 3, 8], [6, 5, 1, 2]

Q2(c): n = 2

Consider the following dataset:

k-means:

1. Randomly initialise n cluster centroids

- e.g. let 2 initial seeds be ID = 3, 4

2. Iteratively assign each data point to the nearest centroid

3. Update centroids as the mean of assigned points until convergence

ID	Color	Weight (g)	
1	Red	1021.2	
2	Red	1027.0	
3	Red	1012.5	A
4	Blue	1010.4	B
5	Blue	1019.5	
6	Gold	1016.4	
7	Gold	995.4	
8	Gold	1012.8	

- A: (1021.2 + 1027.0 + 1012.5 + 1019.5 + 1016.4 + 1012.8) / 6 = 1018.2
- B: (1010.4 + 995.4) / 2 = 1002.9

Q2(c): n = 2

Consider the following dataset:

ID	Color	Weight (g)		
1	Red	1021.2		
2	Red	1027.0		
3	Red	1012.5	A	A
4	Blue	1010.4	B	B
5	Blue	1019.5		
6	Gold	1016.4		
7	Gold	995.4		
8	Gold	1012.8		

k-means:

1. Randomly initialise n cluster centroids

- e.g. let 2 initial seeds be ID = 3, 4

2. Iteratively assign each data point to the nearest centroid

3. Update centroids as the mean of assigned points until convergence

- A: (1021.2 + 1027.0 + 1012.5 + 1019.5 + 1016.4 + 1012.8) / 6 = 1018.2
- B: (1010.4 + 995.4) / 2 = 1002.9

Q2(c): n = 2

Centroids:

- A: 1018.2
- B: 1002.9

- Assignment of values to clusters unchanged
  - Stop iteration

- ID: [4, 7], [1, 2, 3, 5, 6, 8]

Consider the following dataset:

ID	Color	Weight (g)
1	Red	1021.2
2	Red	1027.0
3	Red	1012.5
4	Blue	1010.4
5	Blue	1019.5
6	Gold	1016.4
7	Gold	995.4
8	Gold	1012.8

Q3:

How could the discrete variable be converted to a continuous numeric variable?

Consider the following dataset:

ID	Color	Weight (g)
1	Red	1021.2
2	Red	1027.0
3	Red	1012.5
4	Blue	1010.4
5	Blue	1019.5
6	Gold	1016.4
7	Gold	995.4
8	Gold	1012.8

Q3:

How could the discrete variable be converted to a continuous numeric variable?

Consider the following dataset:

ID	Color	Weight (g)
1	Red	1021.2
2	Red	1027.0
3	Red	1012.5
4	Blue	1010.4
5	Blue	1019.5
6	Gold	1016.4
7	Gold	995.4
8	Gold	1012.8

1. **Convert names → numbers**:
    - {'Red', 'Blue', 'Gold'} → {0, 1, 2}
2. **One Hot Encoding** (OHE):
    - Create new boolean attributes for every attribute value
    - "Red": [1,1,1,0,0,0,0,0], "Blue": [0,0,0,1,1,0,0,0], "Gold": [0,0,0,0,0,1,1,1]

Naive Bayes

- Task: classify an instance T = < x_1, x_2, … , x_n > into one of the possible classes $c_j \in C$

$$\hat{c} = \arg\max_{c_j \in C} P(c_j | x_1, x_2, \dots, x_n)$$

Choose class w/ highest prob. given attributes x_1 to x_n

$$= \arg\max_{c_j \in C} \frac{P(x_1, x_2, \dots, x_n | c_j) P(c_j)}{\cancel{P(x_1, x_2, \dots, x_n)}}$$

Bayes Rule

Same for all $c_j \in C$, so we can ignore it

$$= \arg\max_{c_j \in C} P(x_1, x_2, \dots, x_n | c_j) P(c_j)$$

?

“Naive” = Assume all attributes are independent

$$P(x_1, x_2, \dots, x_n | c_j) \approx P(x_1|c_j)P(x_2|c_j) \dots P(x_n|c_j)$$
$$= \prod_i P(x_i|c_j)$$

$$\hat{c} = \underset{c_j \in C}{\arg\max}\, P(c_j | x_1, x_2, \ldots, x_n)$$

$$= \underset{c_j \in C}{\arg\max}\, \boxed{P(c_j)} \boxed{\prod_i P(x_i | c_j)}$$

Bayesian Prior Independence assumption

= Probability of each class prior receiving additional info.

= frequency of each class in training data

Case Study

$$\arg\max_{c_j \in C} P(c_j) \prod_i P(x_i|c_j)$$

What do we need to calculate?

Given the following dataset, build a Naive Bayes model to predict the label "Play."

ID	Outlook	Temp	Humid	Wind	Play
A	S	H	N	F	N
B	S	H	H	T	N
C	O	H	H	F	Y
D	R	M	H	F	Y
E	R	C	N	F	Y
F	R	C	N	T	N

1. **Priors: P(c_j)**
2. **Conditional probabilities: P(x_i|c_j)**

Case Study

$$\arg\max_{c_j \in C} P(c_j) \prod_i P(x_i|c_j)$$

Step 1: Calculate Prior

Given the following dataset, build a Naive Bayes model to predict the label "Play."

ID	Outlook	Temp	Humid	Wind	Play
A	S	H	N	F	N
B	S	H	H	T	N
C	O	H	H	F	Y
D	R	M	H	F	Y
E	R	C	N	F	Y
F	R	C	N	T	N

= Probability of each class prior receiving additional info.
= frequency of each class in training data

- P(play=N) = 1/2
- P(play=Y) = 1/2

Case Study

$$\arg\max_{c_j \in C} P(c_j) \prod_i P(x_i|c_j)$$

Step 2: Calculate Conditional

Given the following dataset, build a Naive Bayes model to predict the label "Play."

ID	Outlook	Temp	Humid	Wind	Play
A	S	H	N	F	N
B	S	H	H	T	N
C	O	H	H	F	Y
D	R	M	H	F	Y
E	R	C	N	F	Y
F	R	C	N	T	N

- P(outlook=S | play = N) = 2/3
- P(outlook=O | play = N) = 0
- P(outlook=R | play = N) = 1/3
- P(outlook=S | play = Y) = 0
- P(outlook=O | play = Y) = 1/3
- P(outlook=R | play = Y) = 2/3

Repeat for all other attributes x

Case Study

$$\arg\max_{c_j \in C} P(c_j) \prod_i P(x_i | c_j)$$

Step 2: Calculate Conditional

- P(outlook=S | play = N) = 2/3
- P(outlook=O | play = N) = 0
- P(outlook=R | play = N) = 1/3
- P(outlook=S | play = Y) = 0
- P(outlook=O | play = Y) = 1/3
- P(outlook=R | play = Y) = 2/3

- P(temp=H | play = N) = 2/3
- P(temp=M | play = N) = 0
- P(temp=C | play = N) = 1/3
- P(temp=H | play = Y) = 1/3
- P(temp=M | play = Y) = 1/3
- P(temp=C | play = Y) = 1/3

- P(humid=N | play = N) = 2/3
- P(humid=H | play = N) = 1/3
- P(humid=N | play = Y) = 1/3
- P(humid=H | play = Y) = 2/3

- P(wind=F | play = N) = 1/3
- P(wind=T | play = N) = 2/3
- P(wind=F | play = Y) = 1
- P(wind=T | play = Y) = 0

Case Study

$$\underset{c_j \in C}{\arg\max}\, P(c_j) \prod_i P(x_i | c_j)$$

Prediction:

- P(c_j | x) = P(**play = N** | **outlook** = ?, **temp** = ?, **humid** = ?, **wind** = ?)
  - P(play = N) * **P(outlook = ?| play = N)** * **P(temp = ?| play = N)** * **P(humid = ? | play = N)** * **P(wind = ? | play = N)**
- P(c_j | x) = P(**play = Y** | **outlook** = ?, **temp** = ?, **humid** = ?, **wind** = ?)
  - P(play = Y) * **P(outlook = ?| play = Y)** * **P(temp = ?| play = Y)** * **P(humid = ? | play = Y)** * **P(wind = ? | play = Y)**

Case Study

Use the model to classify these test instances (? represents missing value):

ID	Outlook	Temp	Humid	Wind	Play
G	O	M	N	T	?
H	?	H	?	F	?

- P(play = N | outlook = O, temp = M, humid = N, wind = T)
  - P(play = N) * P(outlook = O| play = N) * P(temp = M| play = N) * P(humid = N | play = N) * P(wind = T | play = N)

- P(play = Y | outlook = O, temp = M, humid = N, wind = T)
  - P(play = Y) * P(outlook = O| play = Y) * P(temp = M| play = Y) * P(humid = N | play = Y) * P(wind = T | play = Y)

Q4:

Use the model to classify these test instances (? represents missing value):

ID	Outlook	Temp	Humid	Wind	Play
G	O	M	N	T	?
H	?	H	?	F	?

Classify the test instances using:

1. No smoothing
2. Epsilon smoothing
3. Laplace smoothing (alpha = 1)

Case Study

$$\arg\max_{c_j \in C} P(c_j) \prod_i P(x_i | c_j)$$

Step 2: Calculate Conditional

- P(outlook=S | play = N) = 2/3
- P(outlook=O | play = N) = 0
- P(outlook=R | play = N) = 1/3
- P(outlook=S | play = Y) = 0
- P(outlook=O | play = Y) = 1/3
- P(outlook=R | play = Y) = 2/3

- P(temp=H | play = N) = 2/3
- P(temp=M | play = N) = 0
- P(temp=C | play = N) = 1/3
- P(temp=H | play = Y) = 1/3
- P(temp=M | play = Y) = 1/3
- P(temp=C | play = Y) = 1/3

- P(humid=N | play = N) = 2/3
- P(humid=H | play = N) = 1/3
- P(humid=N | play = Y) = 1/3
- P(humid=H | play = Y) = 2/3

- P(wind=F | play = N) = 1/3
- P(wind=T | play = N) = 2/3
- P(wind=F | play = Y) = 1
- P(wind=T | play = Y) = 0

Q4(a): No Smoothing

ID	Outlook	Temp	Humid	Wind	Play
G	O	M	N	T	?
H	?	H	?	F	?

- P(**play = N** | **outlook** = O, **temp** = M, **humid** = N, **wind** = T)
    - = P(play = N) * **P(outlook = O| play = N)** * **P(temp = M| play = N)** * **P(humid = N | play = N)** * P(wind = T | play = N)

    = 1/2 * **0** * **0** * 2/3 * 2/3 = 0

- P(**play = Y** | **outlook** = O, **temp** = M, **humid** = N, **wind** = T)
    - = P(play = Y) * **P(outlook = O| play = Y)** * **P(temp = M| play = Y)** * **P(humid = N | play = Y)** * P(wind = T | play = Y)

    = 1/2 * 1/3 * 1/3 * 1/3 * **0** = 0

→ Both 0 probability, no labels predicted

Q4(a): No Smoothing

ID	Outlook	Temp	Humid	Wind	Play
G	O	M	N	T	?
H	?	H	?	F	?

- P(**play = N** | **outlook** = ?, **temp** = H, **humid** = ?, **wind** = F)

  = P(play = N) * **P(temp = H | play = N)** * **P(wind = F | play = N)**

  = 1/2 * 2/3 * 1/3 = 1/9

- P(**play = Y** | **outlook** = ?, **temp** = H, **humid** = ?, **wind** = F)

  = P(play = Y) * **P(temp = H | play = Y)** * **P(wind = F | play = Y)**

  = 1/2 * 1/3 * 1 = 1/6

→ 1/6 > 1/ 9 → Predict **play = Y**

Q4(b): Epsilon

ID	Outlook	Temp	Humid	Wind	Play
G	O	M	N	T	?
H	?	H	?	F	?

Epsilon smoothing = replace 0 values with epsilon

- P(**play = N** | **outlook** = O, **temp** = M, **humid** = N, **wind** = T)

  = 1/2 * **epsilon** * **epsilon** * 2/3 * 2/3 = 2/9 * epsilon^2

- P(**play = Y** | **outlook** = O, **temp** = M, **humid** = N, **wind** = T)

  = 1/2 * 1/3 * 1/3 * 1/3 * **epsilon** = 1/54 * epsilon

→ Which is larger? Here, we choose **play = Y**. Why?

Q4(b): Epsilon

ID	Outlook	Temp	Humid	Wind	Play
G	O	M	N	T	?
H	?	H	?	F	?

- No zero values → same as no smoothing

Q4(c): Laplace

ID	Outlook	Temp	Humid	Wind	Play
G	O	M	N	T	?
H	?	H	?	F	?

- P(**play = N** | **outlook** = O, **temp** = M, **humid** = N, **wind** = T)
    - = P(play = N) * **P(outlook = O| play = N)** * **P(temp = M| play = N)** * **P(humid = N | play = N)** * P(wind = T | play = N)

  = 1/2 * **(0+1)/(3+3)** * **(0+1)/(3+3)** * (2+1)/(3+2) * (2+1)/(3+2) = 0.005

- P(**play = Y** | **outlook** = O, **temp** = M, **humid** = N, **wind** = T)
    - = P(play = Y) * **P(outlook = O| play = Y)** * **P(temp = M| play = Y)** * **P(humid = N | play = Y)** * P(wind = T | play = Y)

  = 1/2 * (1+1)/(3+3) * (1+1)/(3+3) * (1+1)/(3+2) * **(0+1)/(3+2)** = 0.0044

→ 0.005 > 0.0044 → Predict **play = N**

Unsmoothed:

$$P_i = \frac{x_i}{N}$$

Smoothed:

$$P_i = \frac{x_i + \alpha}{N + \alpha d}$$

d = number of unique values for the attribute

Q4(c): Laplace

ID	Outlook	Temp	Humid	Wind	Play
G	O	M	N	T	?
H	?	H	?	F	?

- P(**play = N** | **outlook** = ?, **temp** = H, **humid** = ?, **wind** = F)

  = P(play = N) * **P(temp = H | play = N)** * **P(wind = F | play = N)**

  = 1/2 * (2+1)/(3+3) * (1+1)/(3+2) = 0.1

- P(**play = Y** | **outlook** = ?, **temp** = H, **humid** = ?, **wind** = F)

  = P(play = Y) * **P(temp = H | play = Y)** * **P(wind = F | play = Y)**

  = 1/2 * (1+1)/(3+3) * (3+1)/(3+2) = 0.13

→ 0.13 > 0.1 → Predict **play = Y**

Unsmoothed:

$$P_i = \frac{x_i}{N}$$

Smoothed:

$$P_i = \frac{x_i + \alpha}{N + \alpha d}$$