Tutorial

Classification (2)

Q1 Using rules as features

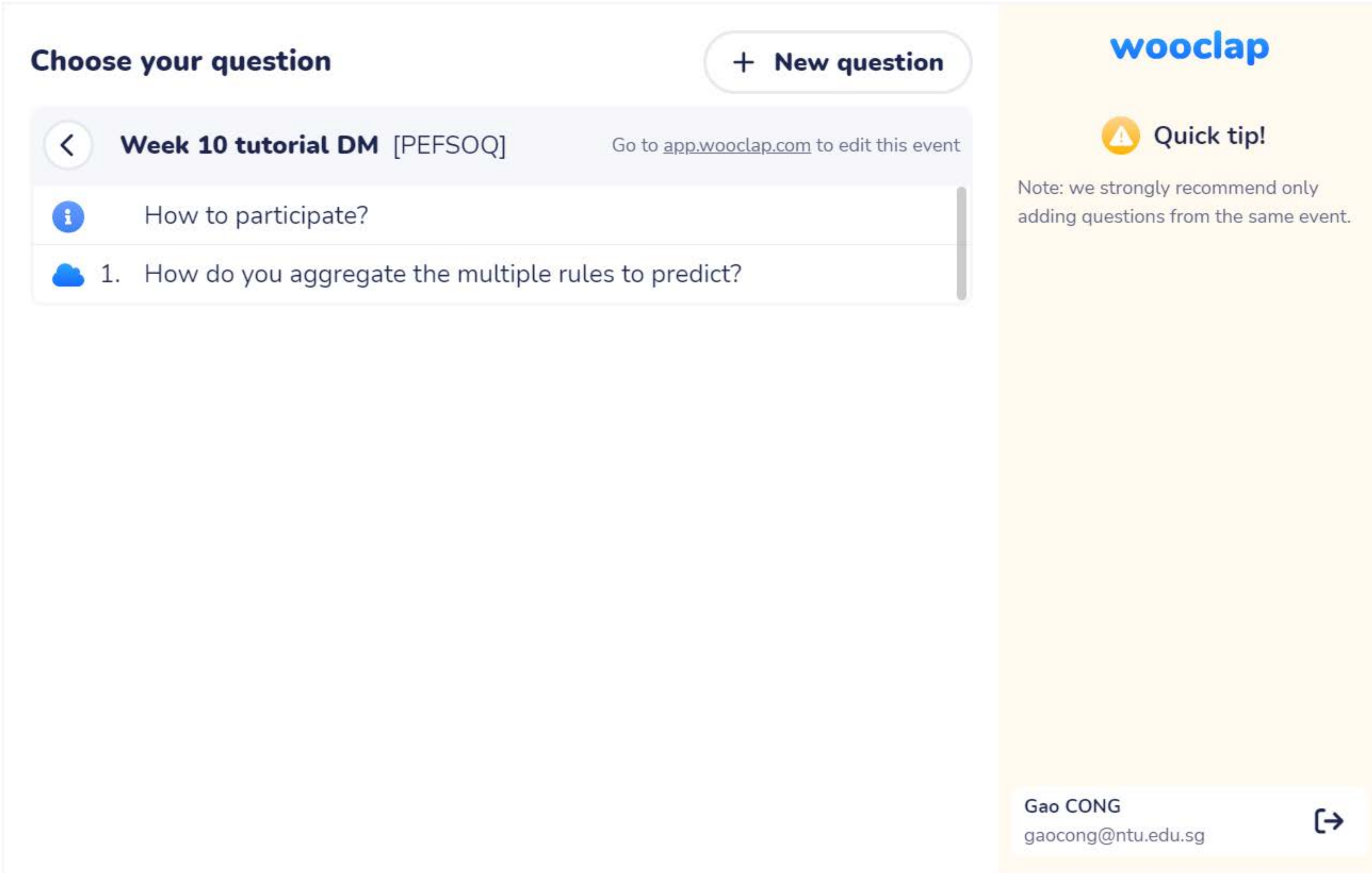
- Most classification methods do not fully explore multi-attribute correlations, e.g., naïve Bayesian, decision trees, rules induction, etc.
- Option 1: This method creates extra attributes to augment the original data by
  - Using the conditional parts of rules
  - Each rule forms a new attribute
  - If a data record satisfies the condition of a rule, the attribute value is 1, and 0 otherwise
- Option 2: use only rules as attributes
  - Throw away the original data

Then use any existing classifier, such as decision tree, NN, SVM

Q2 Give a solution of using frequent sequential patterns to generate rules, and use the generated rules to predict the next item.

- Derive rules from frequent sequential patterns
  - E.g., <PC, camera> -- > Phone (sup: 80%, conf 85%)
- Match the top-5 rules, and use them to predict

How do you aggregate the multiple rules to predict?



Q3difference between bagging and boosting

- Boosting
- An iterative procedure to adaptively change distribution of training data by focusing more on previously misclassified records
  - Initially, all N records are assigned equal weights
  - Unlike bagging, sampling weights may change at the end of boosting round
  - Each classifier normally has different weight for aggregating the final scores.

Q4 using the idea of boosting based CBA.

1. Sample data
2. For each sampled data,
    1. build a CBA classifier
    2. Compute a weight for the CBA classifier
    3. Adjust weight of data
    4. Resample, and go to step 2, until the end of the rounds.

Q5 Class Imbalance Problem

- Lots of classification problems where the classes are skewed (more records from one class than another)
  - Credit card fraud
  - Intrusion detection
  - Defective products in manufacturing assembly line
  - COVID-19 test results on a random sample

- **Key Challenge:**
  - Evaluation measures such as accuracy are not well-suited for imbalanced class

Accuracy

	PREDICTED CLASS	
		Class=Yes	Class=No
ACTUAL CLASS	Class=Yes	a (TP)	b (FN)
	Class=No	c (FP)	d (TN)

- Most widely-used metric:

$$\text{Accuracy} = \frac{a+d}{a+b+c+d} = \frac{TP+TN}{TP+TN+FP+FN}$$

Problem with Accuracy

- Consider a 2-class problem
  - Number of Class NO examples = 990
  - Number of Class YES examples = 10
- If a model predicts everything to be class NO, accuracy is 990/1000 = 99 %
  - This is misleading because this trivial model does not detect any class YES example
  - Detecting the rare class is usually more interesting (e.g., frauds, intrusions, defects, etc)

		PREDICTED CLASS	
		Class=Yes	Class=No
ACTUAL CLASS	Class=Yes	0	10
	Class=No	0	990

Which model is better?

A

	PREDICTED	
	Class=Yes	Class=No
ACTUAL Class=Yes	0	10
ACTUAL Class=No	0	990

Accuracy: 99%

B

	PREDICTED	
	Class=Yes	Class=No
ACTUAL Class=Yes	10	0
ACTUAL Class=No	500	490

Accuracy: 50%

Alternative Measures

	PREDICTED CLASS	
	Class=Yes	Class=No
ACTUAL CLASS Class=Yes	a	b
ACTUAL CLASS Class=No	c	d

$$\text{Precision (p)} = \frac{a}{a+c}$$

$$\text{Recall (r)} = \frac{a}{a+b}$$

$$\text{F-measure (F)} = \frac{2rp}{r+p} = \frac{2a}{2a+b+c}$$

Alternative Measures

	PREDICTED CLASS	
	Class=Yes	Class=No
ACTUAL CLASS Class=Yes	10	0
ACTUAL CLASS Class=No	10	980

$$\text{Precision (p)} = \frac{10}{10+10} = 0.5$$

$$\text{Recall (r)} = \frac{10}{10+0} = 1$$

$$\text{F - measure (F)} = \frac{2*1*0.5}{1+0.5} = 0.62$$

$$\text{Accuracy} = \frac{990}{1000} = 0.99$$

Alternative Measures

	PREDICTED CLASS	
ACTUAL CLASS	Class=Yes	Class=No
Class=Yes	10	0
Class=No	10	980

$$\text{Precision (p)} = \frac{10}{10+10} = 0.5$$

$$\text{Recall (r)} = \frac{10}{10+0} = 1$$

$$\text{F - measure (F)} = \frac{2*1*0.5}{1+0.5} = 0.62$$

$$\text{Accuracy} = \frac{990}{1000} = 0.99$$

	PREDICTED CLASS	
ACTUAL CLASS	Class=Yes	Class=No
Class=Yes	1	9
Class=No	0	990

$$\text{Precision (p)} = \frac{1}{1+0} = 1$$

$$\text{Recall (r)} = \frac{1}{1+9} = 0.1$$

$$\text{F - measure (F)} = \frac{2*0.1*1}{1+0.1} = 0.18$$

$$\text{Accuracy} = \frac{991}{1000} = 0.991$$

Q3 Building Classifiers with Imbalanced Training Set

- Modify the distribution of training data so that rare class is well-represented in training set
  - Undersample the majority class
  - Oversample the rare class

Q4

User based CF of 3 most similar users.

Cosine similarity of [u1, u3..u12] with u2:
[0.372, 0.29, 0.217, 0.527, 0.0, 0.325, 0.198, 0.475, 0.667, 0.487, 0.0]
Rankings of users based on similarity:
[10, 5, 11, 9, 1, 7, 3, 4, 8, 12, 6]
Top 3 users who rated movie 1:
u11, u9, u1 (because u10 and u5 didn't rate movie 1)

Similarity-weighted recommendation:

$$\frac{0.487 * 4 + 0.475 * 5 + 0.372 * 1}{0.487 + 0.475 + 0.372} = 3.519$$

Unweighted recommendation:

$$\frac{4 + 5 + 1}{3} = 3.33$$

Item based CF of 3 most similar items.

Step 1:

Cosine similarity of [i2…i12] with i1:

[0.528, 0.526, 0.285, 0.302, 0.239, 0.470, 0.913, 0.681, 0.533, 0.257, 0.465]

Rankings of items based on similarity:

[8, 9, 10, 2, 3, 7, 12, 5, 4, 11, 6]

Top 3 items that interact with u2:

i10, i3, i4

Similarity weighted recommendation:

$$\frac{0.533 * 2 + 0.526 * 4 + 0.285 * 2}{0.533 + 0.526 + 0.285} = 2.78$$

Unweighted recommendation:

$$\frac{2 + 4 + 2}{3} = 2.67$$