CS412: Introduction to Data Mining **Fall 2015**

Assignment 4

Due: 11/19/2015 11:59pm

General Instruction

- Errata: After the assignment is released, any further corrections of errors or clarifications will be posted at the Errata page at Piazza. Please watch it.
- Feel free to talk to other members of the class while doing the homework. We are more concerned that you learn how to solve the problem than that you solve it entirely on your own. You should, however, write the solution yourself.
- Please use Piazza first if you have questions about the homework. Also feel free to send us e-mails and come to office hours.
- For each question, you should show the necessary calculation steps and reasoning—not only final results. Keep the solution brief and clear.
- For a good balance of cognitive activities, we label each question with an activity type:
  - **L1 (Knowledge)** Definitions, propositions, basic concepts.
  - **L2 (Practice)** Repeating and practicing algorithms/procedures.
  - **L3 (Application)** Critical thinking to apply, analyze, and assess.

Assignment Submission

- Please submit your work before the due time. **We do NOT accept late submission!**
- Please submit your answers electronically via Compass. Contact CITES/TAs if you have technical difficulties in submitting the assignment.
- For this assignment, **typeset** your answers and submit it in a **single PDF file**. **Handwritten answers or hand-drawn pictures are not acceptable**.

1 Constraint pattern mining (10 points)

a. (2', L1) for a set of values S, and a value v, constraint $v \in S$

b. (2', L1) for a set of values S, and a value v, constraint $max(S) \geq v$

c. (2', L1) for a set of values S, and a value v, constraint $max(S) \leq v$

d. (4', L1) for a set of values S, and a value v, constraints $avg(S) \geq v$ and $avg(S) \leq v$

Answer:

a. : not anti-monotone, monotone, and succinct.

b. : not anti-monotone, monotone, and succinct.

c. : anti-monotone, not monotone, and succinct.

d. : both are convertible anti-monotone, convertible monotone, and not succinct.

2 Advanced pattern mining (10 points)

a. (2', L2) **T/F.** Convertible constraints cannot be exploited in an Apriori mining algorithm.

b. (2', L2) **T/F.** In *PrefixSpan*, physical project is not used because of its slow performance.

c. (2', L1) **T/F.** Mining closed frequent graphs only is lossless compression of the graph database.

d. (2', L3) **T/F.** In sequential pattern mining, the number of length-2 candidates generated from x frequent length-1 sequences is $\frac{3}{2}x^2 - \frac{1}{2}x$

e. (2', L3) **T/F.** For nontrivial constraints (not every possible pattern satisfies the constraint), it cannot be monotone and anti-monotone at the same time.

Answer:

a. : False. By properly converting, one can add pruning based on the monotonicity or anti-monotonicity.

b. : False. If the database cannot fit in memory, physical projection should be combined with pseudo-projection which speeds up i/o at a cost of additional computation for projection on the fly.

c. : True. All the frequent graphs can be generated as subgraphs of the closed frequent graphs.

d. : True. $x \times x + \frac{1}{2}x(x-1)$

e. : True. if S satisfies C, by monotone, all S's supersets satisfy C, and thus the complete set satisfy; by anti-monotone, all subsets of the complete set also satisfy C, which contradicts the assumption that C is not trivial.

3 Sequential pattern mining (30 points)

Suppose a toy sequence database D contains three sequences as follows. Let the minimum support be 3.

customer_id	shopping sequence
1	$(bc)(de)f$
2	$bcdef$
3	$(bc)dbegf$

The following questions require you to perform GSP algorithm.

Answer:
The frequent sequential patterns are $\{L_1, L_2, L_3\}$, and candidates are $\{C_1, C_2, C_3\}$. The steps are shown in Table 1.
Using the same DB, apply PrefixSpan to get the same results.

Answer:

4 Decision Trees (18 points)

ID3 is a simple algorithm for decision tree construction using information gain. The steps of the ID3 algorithm are similar to those introduced in the lecture. In particular, ID3 uses information gain to select decision attributes, and each attribute is used at most once in any root-to-leaf path in the decision tree. You will use ID3 to build a decision tree that predicts whether a candidate will be accepted to the PhD program of some University X, given the student's information about GPA, university, publications, and recommendation.

Purpose

- Understand and practice basic decision tree construction, calculation of information gain measures, and classifier evaluation.

Requirements

- Show the calculations for selecting the decision tree attributes and the labels for each leaf.

a. (6', L2) Using the ID3 algorithm to construct a decision tree using the training data in Table 3. When multiple attributes has best information gain, choose the one whose name appears earliest in alphabetical order. When there is a tie for the majority labels, choose `no`. Show the final decision tree, and the calculations to derive that tree.

Answer
See Figure 1.

b. (4', L2) Evaluate your constructed decision tree using the testing data in Table 4 in terms of the precision and recall for the class `yes`. Show your calculations.

Answer
$precision = 1, recall = 2/3.$

c. (4', L2) What is the worst case time complexity of training a decision tree using ID3 on a dataset with n data records and m attributes each having p possible values? Show your analysis.

Table 1: Sequential patterns mined by Generalized Sequential Patterns(GSP)

C1	L1	C2	L2	C3	L3	C4
b	b:3	bb	bd:3	bdf	bdf:3	null
c	c:3	bc	be:3	bef	bef:3	
d	d:3	bd	bf:3	cdf	cdf:3	
e	e:3	be	cd:3	cef	cef:3	
f	f:3	bf	ce:3			
g		cb	cf:3			
		cc	df:3			
		cd	ef:3			
		ce				
		cf				
		db				
		dc				
		dd				
		de				
		df				
		eb				
		ec				
		ed				
		ee				
		ef				
		fb				
		fc				
		fd				
		fe				
		ff				
		(bc)				
		(bd)				
		(be)				
		(bf)				
		(cd)				
		(ce)				
		(cf)				
		(de)				
		(df)				
		(ef)				

Table 2: Sequential patterns mined by prefixSpan

prefix	projected DB	prefix	projected DB	prefix	projected DB
$\langle b \rangle$	(_c)(de)f cdef (_c)dbegf	$\langle bd \rangle$	(_e)f ef begf	$\langle bdf \rangle$	null
		$\langle be \rangle$	f f gf	$\langle bef \rangle$	null
		$\langle bf \rangle$	null		
$\langle c \rangle$	(de)f def dbegf	$\langle cd \rangle$	(_e)f ef begf	$\langle cdf \rangle$	null
		$\langle ce \rangle$	f f gf	$\langle cef \rangle$	null
		$\langle cf \rangle$	null		
$\langle d \rangle$	(_e)f ef begf	$\langle df \rangle$	null		
$\langle e \rangle$	f f gf	$\langle ef \rangle$	null		
$\langle f \rangle$	null				

id	GPA	univ	published	recommendation	accepted
1	4.0	top-10	yes	good	yes
2	4.0	top-10	no	good	yes
3	4.0	top-20	no	normal	yes
4	3.7	top-10	yes	good	yes
5	3.7	top-20	no	good	yes
6	3.7	top-30	yes	good	yes
7	3.7	top-30	no	good	no
8	3.7	top-10	no	good	no
9	3.5	top-20	yes	normal	no
10	3.5	top-10	no	normal	no
11	3.5	top-30	yes	normal	no
12	3.5	top-30	no	good	no

Table 3: Training Data for the decision tree problem

id	GPA	univ	published	recommendation	accepted
1	4.0	top-10	yes	good	yes
2	3.7	top-30	yes	good	yes
3	3.5	top-30	yes	good	yes
4	3.7	top-10	no	good	no
5	3.5	top-30	no	good	no

Table 4: Testing Data for the decision tree problem

Answer

$O(m^2n)$. Each level involves at most n data points, and for each tree node, information gain is computed for at most m attributes, so at most $O(nm)$ time is needed for each level (because we go through the data associated with a node once in order to compute the information gain of each of the m attributes). The tree has at most m levels.

d. (4', L3) Each root-to-leaf path in any decision tree can be converted into a rule, such as a path $A_1 \xrightarrow{=True} A_2 \xrightarrow{=False} class = +1$ can be converted to the rule "If attribute A_i is true and attribute A_2 is false, then the instance has class +1". Please do/answer the following:

1. Generate the rules for each leaf of your constructed decision tree.

Answer

1. If GPA=4.0, then accepted
2. If GPA=3.5 then not accepted
3. If GPA=3.7 and has publication, then accepted
4. If GPA=3.7, has no publication, but in top-20 university, then accepted

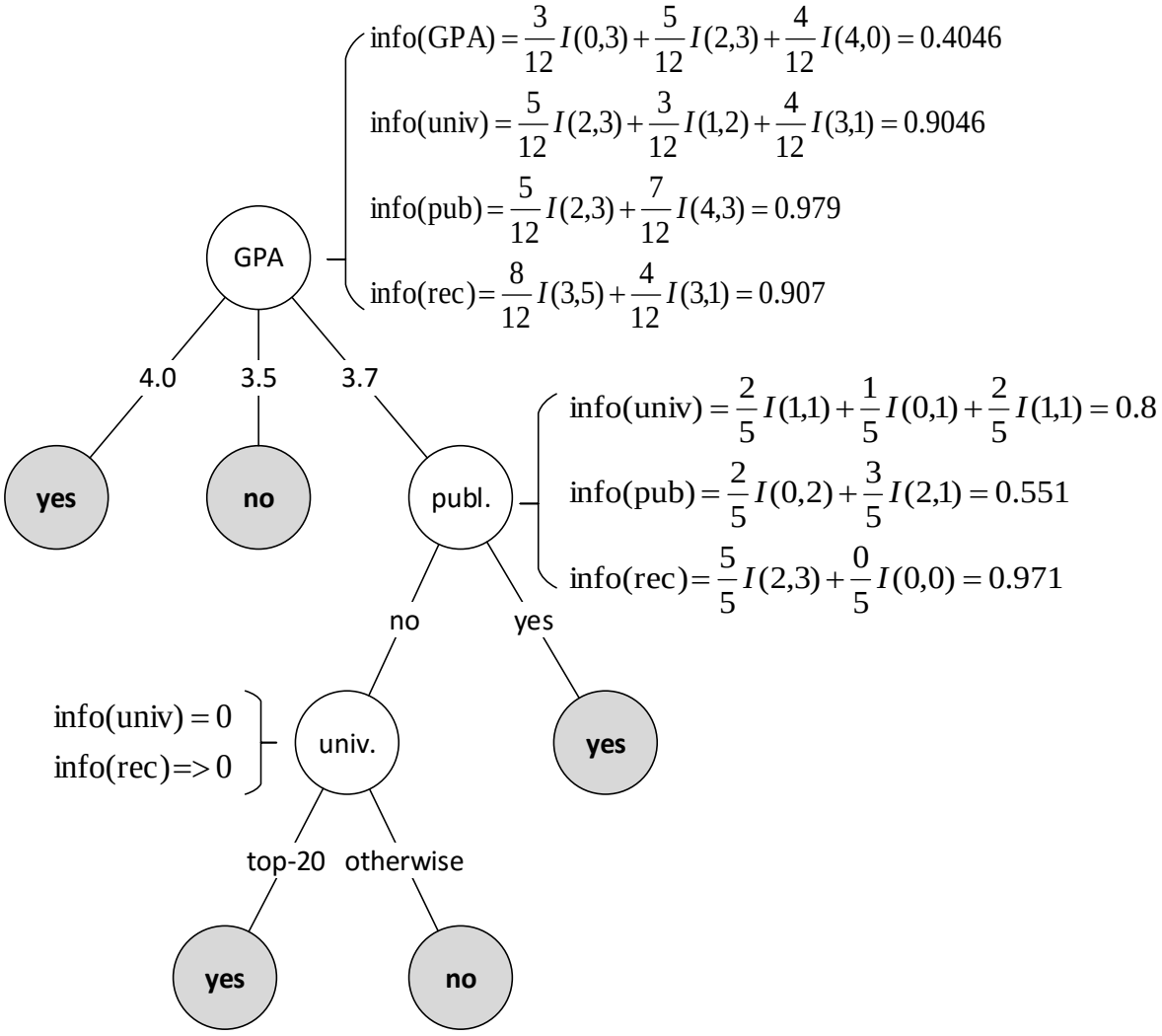

Figure 1: Decision tree by ID3

5. If GPA=3.7, has no publication, and not in top-20 university, then not accepted

(Note: The last two rules might be counterintuitive and perhaps caused by overfitting of the decision tree.)

2. Is it possible to construct a decision tree from a set of rules? Explain your answer.

Answer

Given a set of rules S, construct a decision tree as follows. Select any attribute A that is present every rule in S, and use A as the decision attribute of root r of the decision tree. If one cannot find such an attribute A, or the predict values of A found in S is overlapping, then the construction fails. Otherwise, branch r into branches $r_1, \ldots, r_k$ assuming there are k values for A appearing in the

corresponding rules, and partition the set of rules into disjoint subsets $S_1, \ldots, S_k$. Recursively construct each subtree rooted at r_k with the set of rules S_k.

5 AdaBoost (16 points)

You will be guided through the steps of building an ensemble classifier using AdaBoost. The data points to be classified are given in Table 5. Each classifier in the ensemble will have *one* of the following forms:

- If $x > a$, label +1, else label -1
- If $x \geq a$, label +1, else label -1
- If $y < b$, label +1, else label -1
- If $y \leq b$, label +1, else label -1

where a and b are constants for you to figure out. That is, the hypothesis of the classifier can be represented by a line parallel to the y-axis or the x-axis. While the original AdaBoost algorithm trains each base classifier on *sampled* data points, you will simulate AdaBoost (deterministically) by picking each base classifier given *all* the data points such that the base classifier minimizes the weighted error rate.

id	x	y	**label**
1	1.0	0.5	+1
2	2.2	1.0	+1
3	2.7	2.0	+1
4	0.5	1.5	-1
5	1.2	2.3	-1
6	1.5	2.7	-1

Table 5: Data points for the AdaBoost problem

Purpose

- Understand and practice AdaBoost algorithm by walking through the steps.

Requirements

- Show all the steps and calculations needed to derive each classifier.
- In case of ties when selecting classifiers, pick one that corresponds to a line parallel to the y-axis; if the tie still exists, pick one with minimum a or b.

a. (2', L2) Assume that data weight distribution D_1 in Round 1 is uniform. Find classifier h_1 that has minimum weighted error with data weight distribution D_1. (*Note: see requirements for breaking ties when choosing from equally good classifiers*).

Answer

$h_1 = 2\mathbb{I}_{>1.5}(x) - 1$.

b. (2', L2) What is the weighted error rate of classifier h_1 with data weights D_1?

Answer
$\epsilon_1 = 1/6$.

c. (2', L2) After re-weighting the data according to the results from Round 1, what is the updated data weight distribution D_2 for Round 2? Normalize the weights so that they sum to 1.

Answer
$D_2 = (0.5, 0.1, 0.1, 0.1, 0.1, 0.1)$.

d. (2', L2) Find classifier h_2 for Round 2 that have the minimum weighted error rate for the data weight distribution D_2.

Answer
$h_2 = 2\mathbb{I}_{\leq 1}(y) - 1$
$\epsilon_2 = 0.1$

e. (4', L2) Similar to (c) and (d), compute the weight distribution D_3 and find a classifier h_3 for Round 3.

Answer
$D_3 = (0.5, 0.1, 0.9, 0.1, 0.1, 0.1)/1.8$
$h_3 = 2\mathbb{I}_{\leq 2}(y) - 1$
$\epsilon_3 = 1/18$.

f. (4', L3) What is the ensemble classifier h' that combines h_1, h_2, and h_3? Show h' by plotting the given data points in a 2-D plane and highlighting the regions where points would be classified as $+1$ by h'.

Answer

$h' = sign(\alpha_1 h_1 + \alpha_2 h_2 + \alpha_3 h_3)$, where $\alpha_1 = \log 5, \alpha_2 = \log 9, \alpha_3 = \log 17$. See Figure 2 for the regions where points will be classified as '+1' by h'.

6 Bayes Classifier (16 points)

Using the same training data as in Table 3, your will train a classifier using the Naive Bayes method.

Purpose

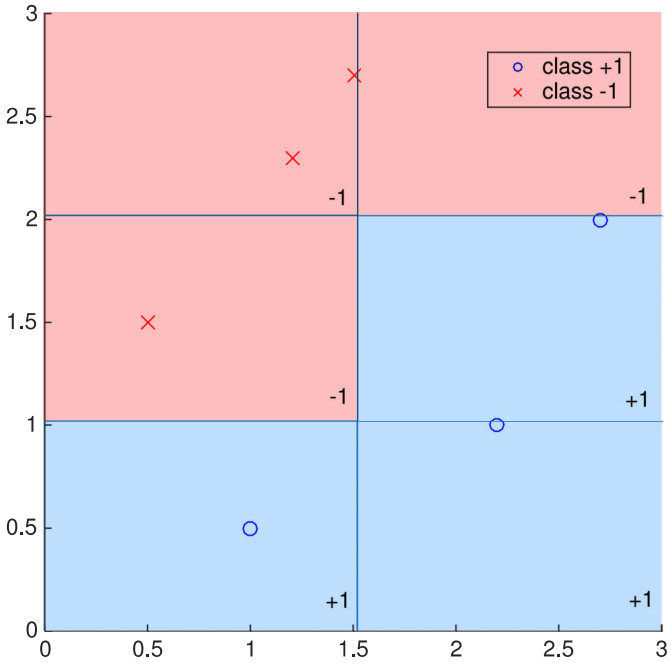

Figure 2: Class regions by emsemble classifier h'.

- Understand and practice the principles of Naive Bayes classifier and its training algorithm; compare the trained classification models to see their pros/cons.

Requirements

- Show the steps and calculations to derive the classifier.
- Show the formulas you used to calculate the results.

a. (1', L2) What is the prior probability of `accepted` being `yes`/`no` estimated from the data?

Answer
$p(\texttt{accepted} = yes) = p(\texttt{accepted} = no) = 0.5$.

b. (3', L2) What is the conditional probability of attribute in `GPA` taking each of the values in $\{4.0, 3.7, 3.5\}$, given `accepted=yes`? Also calculate the conditional probabilities for each of attributes {`university`, `published`, `recommendation`} taking each of its possible values given that `accepted=yes`.

Answer
See Table 6.

		GPA			univ.			publ.		recomm.	
		4.0	*3.7*	*3.5*	*top-10*	*top-20*	*top-30*	*yes*	*no*	*good*	*normal*
acc.	*yes*	1/2	1/2	0	1/2	1/3	1/6	1/2	1/2	5/6	1/6
	no	0	1/3	2/3	1/3	1/6	1/2	1/3	2/3	1/2	1/2

Table 6: Conditional probabilities for Naive Bayes

c. (3', L2) Calculate similar conditional probabilities asked in (b) with the conditions replaced by `accepted=no`.

Answer
See Table 6.

d. (3', L2) Based the results you got from (a)-(c), given a student with attributes (`GPA=3.7, university=top-20, published=yes, recommendation=good`), calculate the probability of the student being accepted. What will the probability become if the student has (`GPA=3.7, university=top-30, publication=no, recommendation=normal`)?

Answer
0.882, 0.111.
The probability can be calculated using the Bayes formula $P(C|X) = P(X|C)P(C)/P(X)$. Note that $P(X)$ is given by $P(X|C)P(C)+P(X|\neg C)P(\neg C)$ and $P(X|C) = \prod_i P(X_i|C)$.

e. (2', L2) Consider a training dataset with n tuples and m attributes, and assume each attributes can take k possible values. What is the time complexity of training a Naive Bayes classifier (for binary prediction)? Show your analysis.

Answer
Scan the table of training data, and for each tuple, update $P(X_i|C)$ in $O(1)$ time for $i = 1 \ldots m$). There are n tuples. So the total time is $O(mn)$.

f. (4', L3) Discuss the pros and cons of the classification models have trained using decision trees and Naive Bayes (Name one pro and one con for each model).

Answer
Decision trees can generate intuitive classification rules, while Naive Bayes cannot. Naive Bayes classifiers not only predict the labels but also give the probabilities of having the labels, which can be interpreted as, for example, confidence with the prediction.

7 Frequent Subgraph Pattern Mining (15 points)

Answer:

- Only H, and $O - H$
- Run the code with support 0.66
- carboxyl group (COOH) is the structure that determins organic component. In most cases, O-H features in inorganic component (though in fact, a H atom tented to be released as $H+$ ion makes it acid, but atom O often co-occurs to take electron from H)