CS4445 Data Mining and Knowledge Discovery in Databases. A Term 2008

Exam 1 - September 26, 2008 – SOLUTIONS

Prof. Carolina Ruiz
Department of Computer Science
Worcester Polytechnic Institute

NAME: Prof. Ruiz

Problem I: **(/10 points)** Data Preprocessing

Problem II: **(/32 points)** Decision Trees

Problem III: **(/35 points)** Classification Rules

Problem IV: **(/33 points)** Association Rules

TOTAL SCORE: (/100 points)

Instructions:

- Show your work and justify your answers
- Use the space provided to write your answers
- Ask in case of doubt

Problem I. Data Preprocessing [10 points]

What is the difference between supervised and unsupervised discretization? Explain decisively.

Solution:

In supervised discretization, the target attribute is used to help determine appropriate bins for the attribute being discretized. In unsupervised discretization, the attribute being discretized is split into bins that are determined without taking the target attribute (if any) under consideration.

Problem II. Decision Trees [32 points]

Consider the following toy dataset adapted from the Iris dataset.

ATTRIBUTES:	POSSIBLE VALUES:
sepallength	{sl-short,sl-med,sl-long}
petallength	{pl-short,pl-med,pl-long}
petalwidth	{pw-short,pw-med,pw-long}
class	{Iris-setosa,Iris-versicolor,Iris-virginica}

id#	sepallength	petallength	petalwidth	class
1	sl-short	pl-short	pw-short	Iris-setosa
2	sl-short	pl-short	pw-short	Iris-setosa
3	sl-short	pl-short	pw-short	Iris-setosa
4	sl-long	pl-med	pw-med	Iris-versicolor
5	sl-long	pl-long	pw-med	Iris-versicolor
6	sl-med	pl-med	pw-med	Iris-versicolor
7	sl-med	pl-med	pw-med	Iris-versicolor
8	sl-med	pl-long	pw-med	Iris-virginica
9	sl-med	pl-long	pw-long	Iris-virginica
10	sl-long	pl-long	pw-long	Iris-virginica

Consider the ID3 algorithm to construct a decision tree for predicting the attribute **class**.

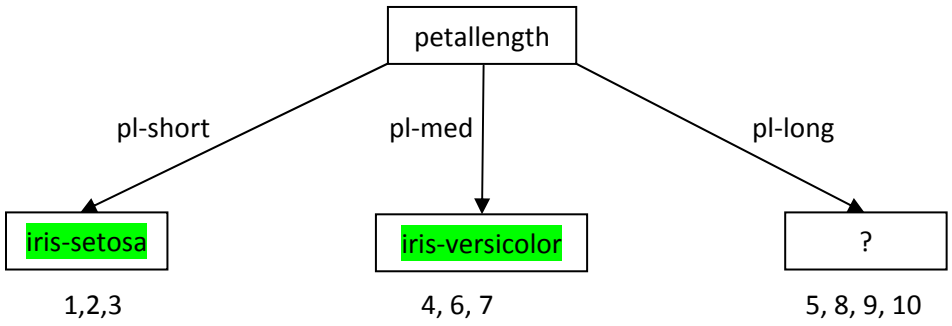
1. Assume that root node of the ID3 tree is **petallength** (you don't need to do entropy calculations to determine the root node. We are telling you that it is petallength).
   [2 points] Depict the root node of the tree with the 3 associated branches.
   [3 points] Include in each branch the **id#s** of the instances that reach that branch.

Solutions: See next page.

2. Starting from this root node, construct the FULL decision tree for this dataset USING THE ID3 ALGORITHM. For your convenience, the logarithms in base 2 of selected values are provided.
   [16 points] Show all the steps of the entropy calculations.
   [5 points] Depict the tree at each stage. At each node state what instances (use **id#s**) are included in the node, what attributes are available to split the node, and which attribute is selected based on entropy.

x	1/2	1/3	2/3	1/4	3/4	1/5	2/5	3/5	1/6	5/6	1/7	2/7	3/7	4/7	1
$\log_2(x)$	-1	-1.5	-0.6	-2	-0.4	-2.3	-1.3	-0.7	-2.5	-0.2	-2.8	-1.8	-1.2	-0.8	0

Solutions:



PETALLENGTH=pl-short: Since instances 1,2, 3 all have class=iris-setosa, the node is homogeneous and its prediction is iris-setosa.

PETALLENGTH=pl-med: Since instances 4,6,7 all have class=iris-versicolor, the node is homogeneous and its prediction is iris-versicolor.

PETALLENGTH=pl-long: It contains instances 5 (iris-versicolor), and 8, 9, 10 (iris-virginica). Since the node it is heterogeneous, we need to split it using one of the two remaining attributes: SEPALLENGTH or PETALWIDTH. We need to calculate the entropy of each of these attributes with respect to instances 5, 8, 9, 10.

id#	**sepallength**	**petallength**	**petalwidth**	**class**
5	sl-long	pl-long	pw-med	Iris-versicolor
8	sl-med	pl-long	pw-med	Iris-virginica
9	sl-med	pl-long	pw-long	Iris-virginica
10	sl-long	pl-long	pw-long	Iris-virginica

CLASS:	iris-setosa	iris-versicolor	iris-virginica

SEPALLENGTH

sl-long: (2/4)*[0 - (1/2)*log(1/2) - (1/2)*log(1/2)]

sl-med: (2/4)*[0 - 0*log(0) - (2/2)*log(2/2)]

= (2/4)*[0 + (1/2) + (1/2)] + (2/4)*[0 + 0 + 0] = ½

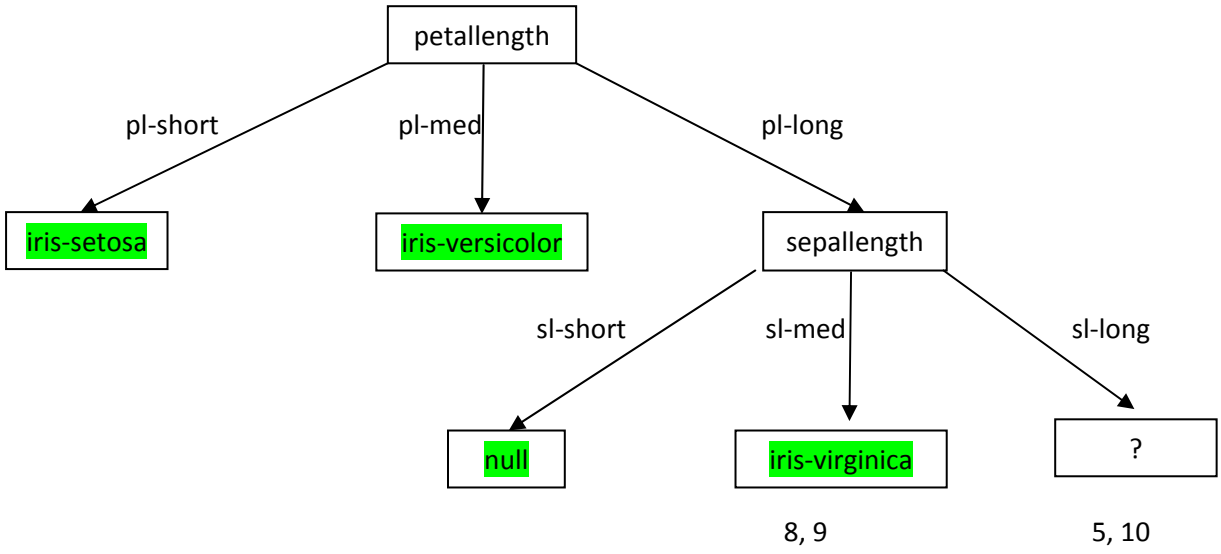
PETALWIDTH:

pw-med: (2/4)*[0 - (1/2)*log(1/2) - (1/2)*log(1/2)]

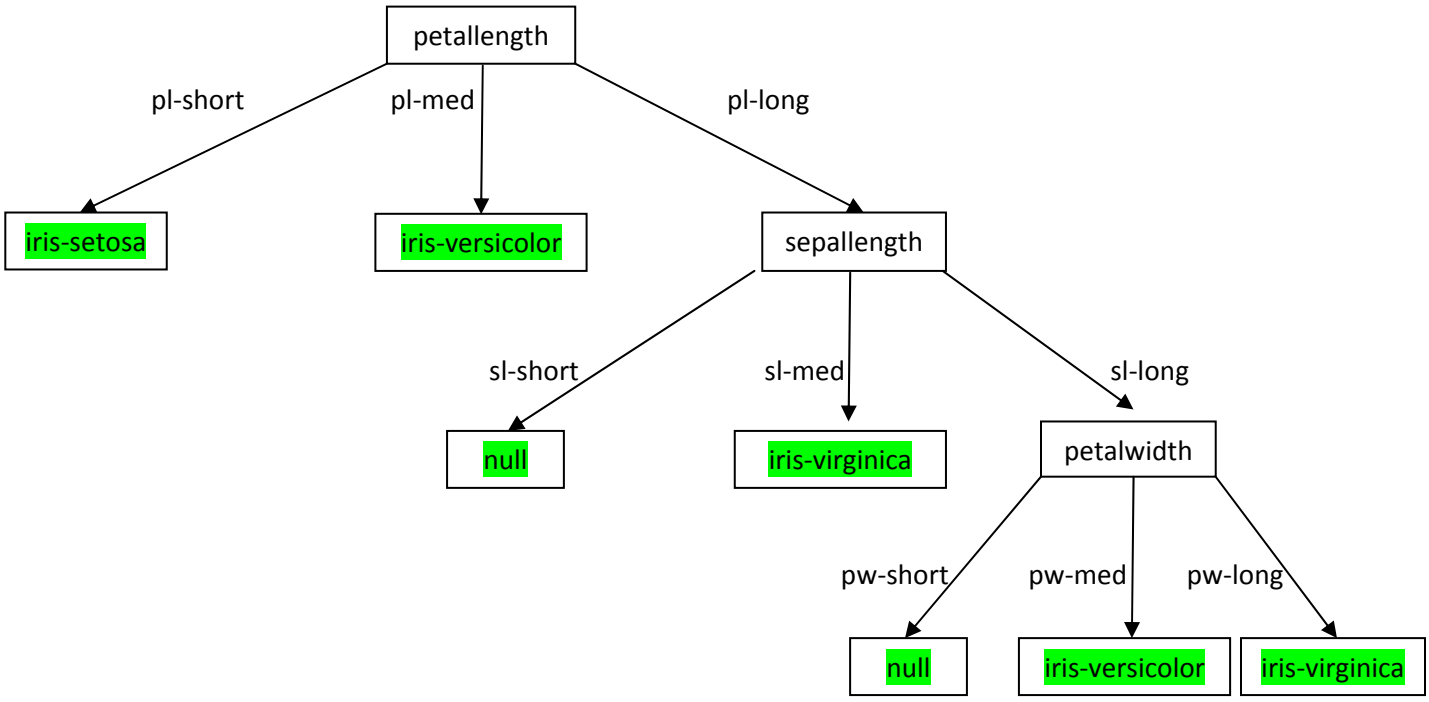
pw-long: (2/4)*[0 - 0*log(0) - (2/2)*log(2/2)]

= (2/4)*[0 + (1/2) + (1/2)] + (2/4)*[0 + 0 + 0] = ½

Since both attributes have the same entropy with respect to these 4 instances, then we can select either one of them. Let's choose SEPALLENGTH. Hence, the tree will look like:



The lower right-most node of the tree is still heterogeneous: 5 (iris-versicolor) and 10 (iris-virginica). So we need to split it. Since we have only one remaining attribute, PETALWIDTH, we'll use it to split this node. The final tree is depicted below.



3. **[6 points]** (This is a general question, independent from the dataset above.) Name three differences between ID3 and J4.8.

Solutions:

- **J4.8 can handle numeric predicting attributes, and ID3 cannot.**
- **J4.8 can handle missing values in predicting attributes, and ID3 cannot.**
- **J4.8 can use pruning, and ID3 cannot.**
- **ID3 trees do not repeat attributes along a branch, but J4.8 may reuse (numeric) attributes along a branch (because of the binary splits).**

Problem III. Classification Rules [35 points].

Consider the weather.nominal dataset:

ATTRIBUTES:	POSSIBLE VALUES:
outlook	{sunny, overcast, rainy}
temperature	{hot, mild, cool}
humidity	{high, normal}
windy	{TRUE, FALSE}
play	{no, yes}

Id#	Outlook	Temperature	Humidity	Windy	Play
1	sunny	hot	high	FALSE	no
2	sunny	hot	high	TRUE	no
3	overcast	hot	high	FALSE	yes
4	rainy	mild	high	FALSE	yes
5	rainy	cool	normal	FALSE	yes
6	rainy	cool	normal	TRUE	no
7	overcast	cool	normal	TRUE	yes
8	sunny	mild	high	FALSE	no
9	sunny	cool	normal	FALSE	yes
10	rainy	mild	normal	FALSE	yes
11	sunny	mild	normal	TRUE	yes
12	overcast	mild	high	TRUE	yes
13	overcast	hot	normal	FALSE	yes
14	rainy	mild	high	TRUE	no

1. Let's use Prism to generate rules from this dataset. Assume that Prism has generated all the rules for **Play=no** (listed below), has generated the first rule for **Play=yes**, and is in the process of generating the second rule for **Play=yes**. Your job is to complete the construction of that second rule for **Play=yes**.

   All Prism rules for **Play=no**:

   If outlook = sunny and humidity = high then no
   If outlook = rainy and windy = TRUE then no

   Prism rules for **Play=yes** so far:

   If outlook = overcast then yes
   If humidity = normal and **?** then yes

Assume that Prism is about to list candidate conditions to fill in the "**?**" in this second rule for **Play=yes**: If humidity = normal and **?** then yes

a. **[5 points]** Circle on the table above the data instances that are under consideration at this point of the execution of Prism during the selection of the 2nd condition of this rule.

   Solution: Instances # 5, 6, 9, 10, 11 are under consideration when the second condition of the second rule for Play=yes is about to be selected.

b. **[5 points]** In front of each of the data instances that you didn't circle above, explain briefly why that data instance wasn't circled.

   Solution:

   - **Instances # 3, 7, 12, and 13 (highlighted in gray in the table above) were eliminated since they are explained by the first rule for Play=yes.**
   - **Instances # 1, 2, 4, 8, and 14 (highlighted in red in the table above) were eliminated since they do not satisfy the first condition of the current rule (namely, "humidity=normal").**

c. **[5 points]** List below **all** the candidate conditions (attribute-value pairs) that need to be considered to replace the "**?**" in this second rule.

d. **[5 points]** For each of these candidate conditions include its **p/t** ratio.

Candidate Conditions	**p/t Ratio**
Solution:	
Outlook=sunny	**2/2**
Outlook=rainy	**2/3**
Temperature=mild	**2/2**
Temperature=cool	**2/3**
Windy=FALSE	**3/3**
Windy=TRUE	**1/2**

e. **[2 points]** Which candidate condition would Prism select to add to the rule? Add that condition to the rule below:

   If humidity = normal and **____ Windy=FALSE _____** then yes

   because this is the condition with the highest p among those with the highest p/t ratio

f. **[1 points]** Will Prism add a third condition to this rule? Why or why not?

Solution: **No, since the rule is already perfect (p/t ratio = 3/3 = 100%)**

g. **[2 points]** Will Prism construct more rules for **Play=yes**? Why or why not?

Solution: **Yes, since there are still data instances (namely id# 4 and 11) with Play=yes that are not explained by the first two Play=yes rules constructed.**

2. **[6 points]** (This is a general question, independent from the dataset above.) Name three differences between the set of rules generated from an ID3 tree and the set of rules generated by Prism over a dataset.

Possible Solutions: **(Some of the differences below are taken from the students' exam answers.)**

PRISM RULES	**ID3 RULES**
• **Perfect rules**	**Not necessarily perfect rules**
• **Rules do not necessarily share attributes**	**All rules share the root attribute**
• **Not all values of an attribute appear in the set of rules**	**If an attribute appears in a rule, then all of its values appear in the set of rules**
• **Constructed in a particular order**	**Not constructed in a particular order**
• **Intended to be used in a particular order**	**Not intended to be used in a particular order**
• **Rules can conflict**	**Rules do not conflict as they come from different branches of the ID3 tree**
• **Constructed using p/t ratio**	**Constructed using entropy metric**
• **Constructed using an iterative covering procedure**	**Constructed using a recursive procedure**

3. **[4 points]** (This is a general question, independent from the dataset above.) Name two differences between the set of rules generated by Prism and the set of (regular, not classification) association rules generated by Apriori over a dataset.

Possible Solutions: (Some of the differences below are taken from the students' exams)

PRISM RULES	**ASSOCIATION RULES**
• **Intended for classification**	**Not intended for classification**
• **Perfect rules**	**The confidence of a rule might be < 100%**
• **The right-hand side of the rule contains only one attribute-value pair, and that attribute must be the target attribute**	**The right-hand side of the rule may contain several attribute-value pairs, and none of them is considered a target attribute**
• **Constructed using p/t ratio**	**Constructed using min. support and min. confidence thresholds.**

Problem IV. Association Rules [33 points].

1. **[21 points]** Assume that the Apriori algorithm is used to generate association rules from a dataset of transactions. Assume also that the **<u>complete list of frequent 4-itemsets</u>** (i.e., all itemsets of size 4 that have enough support) that Apriori has generated is:

Level 4:

{a, b, c, d}

{a, b, c, e}

{a, b, d, e}

{a, c, d, e}

{b, c, d, e}

{c, d, e, f}

{c, d, e, g}

Consider now the 5-itemsets {a, b, c, d, e}, {b, c, d, e, f}, and {c, d, e, f, g}. Answer the following questions assuming that the **join** and the **prune** conditions are used to generate itemsets in Level 5:

	Itemset {a, b, c, d, e}	Itemset {b, c, d, e, f}	Itemset {c, d, e, f, g}
[2 points each] Will this itemset be generated as a candidate 5-itemset using the **join** condition? Yes? No? Maybe? <u>Justify</u> your answer.	**YES. This itemset is the joint of two itemsets in Level 4: {<u>a, b, c</u>, d} and {<u>a, b, c</u>, e}, which satisfy the join condition as they start with the same n-1 (i.e., 3) same items.**	**NO. Due to the join condition, the only way to generate this itemset would be from: {<u>b, c, d</u>, e} and {<u>b, c, d</u>, f}. But {b, c, d, f} does not belong to Level 4.**	**YES. This itemset is the joint of two itemsets in Level 4: {<u>c, d, e</u>, f} and {<u>c, d, e</u>, g}, which satisfy the join condition as they start with the same n-1 (i.e., 3) same items.**
[2 points each] If your answer above for the **join** condition was yes, will this itemset be eliminated by the **prune** condition? Yes? No? Maybe? <u>Justify</u> your answer.	**NO. All the subsets of this itemset appear in Level 4 and are therefore frequent.**	**[No. This itemset is not even generated, let alone pruned.]**	**YES. One or more of its subsets are not frequent since they do not appear in Level 4 (e.g., {d, e, f, g}, and {c, e, f, g}).**
[3 points each] Is this itemset frequent (i.e., does it have enough support)? Yes? No? Maybe? <u>Justify</u> your answer.	**MAYBE. The fact that all its subsets are frequent doesn't guarantee that the itemset is frequent. We'd need to count its support to make sure.**	**NO. Since some of its subsets (e.g., {b, c, d, f}) are not frequent (they do not appear in Level 4) by the Apriori principle, this itemset cannot be frequent.**	**NO. Since some of its subsets (e.g., {d, e, f, g}) are not frequent (they do not appear in Level 4) by the Apriori principle, this itemset cannot be frequent.**

2. **[12 points]** Let X and Y be itemsets in a dataset of transactions. Explain why/how each of the following metrics of an association rule X → Y is useful in the analysis of the dataset.

   a. **Support:**
      i. **[2 points]** Define support(X → Y) =

      Solution: support(X → Y) = P(X & Y): that is, the probability that X and Y appear together in a transaction/instance of the dataset.

      ii. **[2 points]** Why/how is the support of a rule useful information?

      Solution: The support of a rule measures the "coverage" of the rule. This is useful information as it tells us how often X and Y appear together in the dataset transactions/instances.

   b. Confidence:
      i. **[2 points]** Define confidence(X → Y) =

      Solution: confidence(X → Y) = P(Y|X) = P(X & Y)/P(X): that is, the probability that Y appears in a transaction/instance of the dataset in which X occurs.

      ii. **[2 points]** Why/how is the confidence of a rule useful information?

      Solution: The confidence of a rule tells how likely Y is given X. It gives an indication of the "accuracy" of the rule, or in other words, the "strength" of the association between X and Y.

   c. **[4 points] Lift:** lift(X → Y) = P(Y|X)/P(Y) : that is, the ratio between the conditional probability of Y given X, and the probability of Y. What would it mean for X and Y if lift(X → Y) = 1? If lift(X → Y) > 1? If lift(X → Y) < 1? How would this information be useful for the analysis of the dataset?

   Solution:

   - **If lift(X → Y) = P(Y|X)/P(Y) = 1, then P(Y|X) = P(Y). That is, X doesn't make Y more or less likely (than Y by itself). Since P(Y|X) = P(X), then Y is independent from X.**
   - **If lift(X → Y) = P(Y|X)/P(Y) > 1, then P(Y|X) > P(Y). That is, X makes Y more likely (than Y by itself).**
   - **If lift(X → Y) = P(Y|X)/P(Y) < 1, then P(Y|X) < P(Y). That is, X makes Y less likely (than Y by itself).**