CS490D Spring 2004 Midterm Solutions, March 12, 2004
Prof. Chris Clifton

Time will be tight. If you spend more than the recommended time on any question, **go on to the next one**. If you can't answer it in the recommended time, you are either going in to too much detail or the question is material you don't know too well. You can skip one or two parts and still demonstrate what I believe to be an A-level understanding of the material.

Note: It is okay to use abbreviations in your answers, as long as the abbreviations are unambiguous and reasonably obvious.

Grading expectation (initial): A student getting 25 of the available 31 is on track to an A in the course. 19 and above is likely to be a B. Below 15, we should talk (you may not be on track to get a C.)

During the exam, you will make use of the following table, a hypothetical employment statistics record for recent graduates.

ID	*major*	*avg. project score*	*avg. exam score*	*co-op?*	*employed?*	*salary*
1	Computer	87	75	Y	Y	60,000
2	History	?	92	N	N	?
3	Computer	77	95	N	Y	50,000
4	Engineering	97	65	N	N	0
5	Engineering	84	75	Y	Y	40,000

. . .

1 Decision Tree

You have been asked to build a decision tree to predict if someone will be employed following graduation. The are questions about the process you will go through.

1.1 Data Preparation (5 minutes, 5 points)

For each of the following tasks, describe (briefly) what, if anything, you would need to do to the above dataset before using a decision tree learning algorithm.

- Data cleaning

  Missing values need to be handled. Missing project scores could well be not applicable, missing salaries are likely 0. Requires domain knowledge to determine.

  Scoring: 1 point each for noting missing values are an issue, giving an idea how to fix them.

- Data integration

  No problems to be dealt with.

  Scoring: 1/2 point for recognizing no problem (or a good problem if you found one).

- Data transformation

  No problems to be dealt with.

  Scoring: 1/2 point for recognizing no problem (or a good problem if you found one).

- Data reduction

  Drop salary, as it is dependent on the class variable and would not be available for prediction. Drop ID, as it is not really relevant.

  Scoring: 1/2 point each for what to drop, reasoning.

- Data discretization

  Project and exam score need to be discretized. Likely use equi-width binning, assuming capabilities reflected linearly by change in score.

  Scoring: 1 point each for noting problem, describing appropriate discretization method.

Note that this adds up to more than 5 - your score is limited to five, but you don't need a perfect score on all parts to get it.

1.2 Tree building (5 minutes, 5 points)

Discuss which attribute would be picked for the first split by the ID3 algorithm. Describe or set up the calculations, but you don't need to work out the details. An intuitive discussion is fine.

Calculate information gain split on each of project, exam, major, co-op, employed.

$$\begin{aligned}
I(all) &= -\sum_{i=Y,N} \frac{\{3,2\}}{5} \log_2 \frac{\{3,2\}}{5} \\
E(major) &= \sum_{i=C,H,E} \frac{\{2,1,2\}}{5} I(\{C,H,E\}) \\
I(C) &= -\sum_{i=Y,N} \frac{\{2,0\}}{2} \log_2 \frac{\{2,0\}}{2} \\
Gain(major) &= I(all) - E(major)
\end{aligned}$$

Repeat for others. Intuitively, I expect project would be best, depending on binning strategy, as it could give 0 entropy for each depending on how missing values are handled.

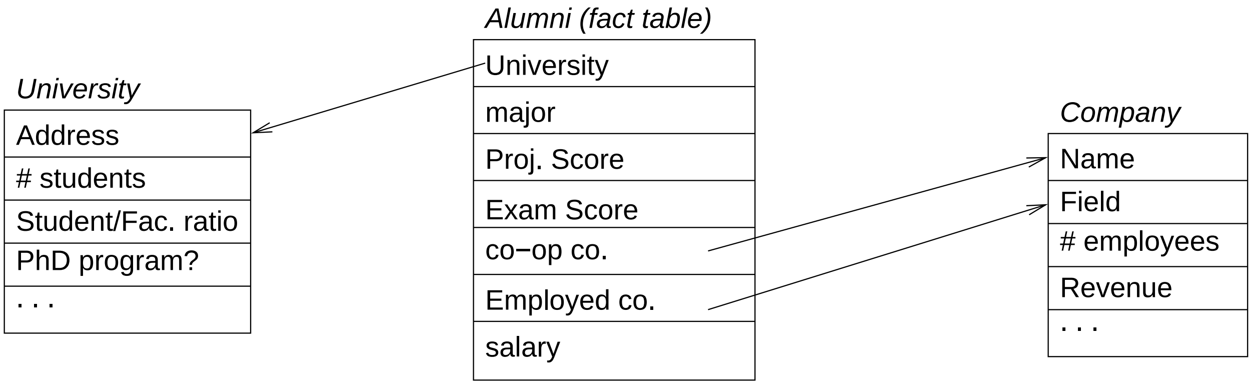
Scoring: 1 point for knowing what needs to be done, 1-2 for setting up equation, 1 point for intuition, 1-2 points for good reasoning.

Extra credit (several minutes, 1 point): Work out the Information Gain for the chosen attribute. **Don't do this unless you've completed the rest of the exam.**

2 Data Warehousing

2.1 Star Schema (5 minutes, 5 points)

Diagram a star schema containing employment statistics data for recent graduates. Based your ideas on the table given at the beginning of the exam, but think of other information that might be relevant. In particular, assume that the warehouse contains data for multiple Universities as well as multiple majors.



Scoring: 1 point for fact table, 1-2 for getting correct values, 1 for each detail table and correct linkage.

2.2 Data Cube (3 minutes, 4 points)

Describe how a data cube could be used to determine:

1. Which University's graduates earn the highest salaries?

   Roll up major, co-op, employment and aggregate for average (or maximum) salary.

   Scoring: 1 point for rolling up, 1 for aggregating.

2. Which University is the best place to study computers?

   Roll up on co-op, employment, select on computers, average salary.

   Scoring: One for some sort of appropriate rolling up and computation, one for selection. Could change depending on how you interpret "best".

3 k-Nearest Neighbor Classification (5 minutes, 4 points)

Could you use a k-Nearest Neighbor Classifier with this dataset? If so, give the 3 nearest neighbors and resulting class for the following instance (explain your reasoning):

ID	major	project score	exam score	co-op?	employed?	salary
6	History	86	74	Y	?	27,000

Closest: 1, 5, and 2 (assuming co-op and major equally weighted), giving (majority) class of employed. *Scoring: 1-2 points for picking closest, 1 for class, 1-2 for reasoning why they are closest (i.e., what you used for distance.)*

If not, explain why not.

Need distances function for k-NN, and it isn't defined for major or co-op.

Scoring: 2 points for noting you need a distance function, 2 for noting why you don't have one.

4 Association Rules (10 minutes, 8 points)

Assume that the "Exam Score" and "Project Score" are discretized by binning into equi-width bins of width 10, i.e., 70-79 go into bin 70. What is the longest association rule you can find (most attributes) with support at least 25% and confidence 100%? For full credit, show the process using an algorithm you've learned.

Using a-priori: C, E, P8, E7, E9, Cp, Em frequent.

2-Candidates: C P8, E P8, C E7, E E7, C E9, E E9, C Cp, E Cp, C Em, E Em, P8 E7, P8 E9, P8 C, P8 Em, E7 C, E7 Em, E9 C, E9 Em, Em Cp.
2-Frequent: C Em, P8 E7, P8 Cp, P9 Em, E7 Cp, E7 Em, Cp Em.
3-Candidates: P8 E7 Em, P8 E7 Cp, P8 Em Cp, E7 Cp Em.
3-Frequent: Same as 3-Candidates.
4-Candidates: P8 E7 Cp Em (all with frequency 2).

For 100% confidence, the left side can occur no more than the right. This is true for:
P8 E7 Em ⇒ Cp
E7 Cp Em ⇒ P8
P8 E7 Cp ⇒ Em
P8 Cp Em ⇒ E7

Scoring: 1 point for getting a frequent rule, 1 for a rule involving 4 attributes, 1 for high confidence, 1 for all such rules, 1-4 for showing understanding of an algorithm.