CSC 480 Sections 901,910 Fall 2018

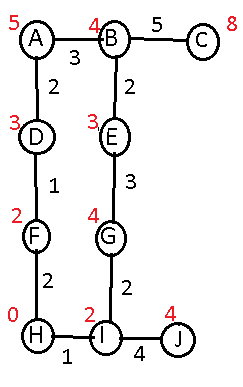
Final Exam

Name:

Directions: This is a take-home exam. It is open book and notes, and computer use is allowed allowed. Please write your answers to the exam questions in this document, and then upload the document by June 8. The exam is worth 25% of your overall grade; therefore it will be graded on a scale of 0 to 25.

Problems

1. (5 points) Consider the graph below. The start state is A, and the goal is H. Red numbers represent the cost estimate from a given vertex to the goal state. Black numbers are actual costs (weights) associated with an edge. Which states must be visited by A\* as it finds the cheapest path to the goal?



|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | # | g | h | f | Pred |
| A | 1 | 0 | 5 | 5 | - |
| B |  | 3 | 4 | 7 | A |
| C |  |  |  |  |  |
| D | 2 | 2 | 3 | 5 | A |
| E |  |  |  |  |  |
| F | 3 | 3 | 2 | 5 | D |
| G |  |  |  |  |  |
| H | 4 (Goal) | 5 | 0 | 5 | F |
| I |  |  |  |  |  |

Working backwards, path is H – F – D – A so the cheapest path from start state A to end state H is:

A – D – F - H

1. (6 points) Consider the following first-order predicate calculus statements, which have already been written in Skolemized clause form.

Write all pairsof clauses below that can resolve (there are 6 of them). If a pair resolves, give the resulting substitution given by unification along with the resolution. **Note**: I am not asking you to give a complete proof, just the result of one resolution at a time.

* 1. inst(sk0(?a), human) v mortal(sk0(?a))
  2. ~inst(Socrates, human)
  3. ~mortal(?b)
  4. ~inst(?c,?d)
  5. inst(Socrates, ?e)
  6. ~inst(?f, human) v ~married(spouse-of(?f)) v single(?f)
  7. married(spouse-of(Socrates))

a. inst(sk0(?a), human) v mortal(sk0(?a))

f. ~inst(?f, human) v ~married(spouse-of(?f)) v single(?f)

resolve to:

(h) mortal(sk0(?a)) v ~married(spouse-of(sk0(?a)) v single(sk0(?a)) // ?f=sk0(?a)

h. mortal(sk0(?a)) v ~married(spouse-of(sk0(?a)) v single(sk0(?a))

g. married(spouse-of(Socrates))

resolve to:

(i) mortal(Socrates) v single(Socrates) //sk0(?a)=Socrates

i. mortal(Socrates) v single(Socrates)

c. ~mortal(?b)

resolve to:

(j) single(Socrates) //?b=Socrates

d. ~inst(?c,?d)

e. inst(Socrates, ?e)

resolve to:

(k) Box //?c=Socrates; ?d=?e

a. inst(sk0(?a), human) v mortal(sk0(?a))

b. ~inst(Socrates, human)

resolve to:

(l) mortal(Socrates) //sk0(?a)=Socrates (also from above)

l. mortal(Socrates)

c. ~mortal(?b)

resolve to:

(m) Box //?b=Socrates

1. (3 points) Say we want to prove that unicorns don’t exist. Here is the theorem:

A x ~unicorn(x)

If we would like to prove that unicorns don’t exist using contradiction and resolution, what is the proper negation of this theorem? Write your answer in clause form. Note: You are not writing a proof in this problem, but merely stating the negation of what you would like to prove.

Ax ~unicorn(x)

Negative:

~Ax ~unicorn(x)

Ex ~ ~ unicorn(x)

Ex unicorn(x)

unicorn(sk0) //Existential quantifier and 1 variable so use skolem term

1. (5 points) You are driving the speed limit on a very fast highway. 90% of drivers on this road exceed the speed limit by 20 MPH or more. 5% speed by less than 20 MPH over the limit, and 5% don’t speed. A car speeds past you just as you reach a speed trap, at which it is known that 70% of drivers who are going 20 MPH or more or the limit are pulled over, and 20% of cars who are going 1-19.999 MPH over the limit are pulled over. The cops don’t pull over non-speeders. You were driving at exactly the speed limit when a car passes you. What is the probability that the car which is passing you will be pulled over?

PO = Pulled Over

GT20 = Greater than 20 MPH over the speed limit

LT20 = Less than 20 MPH over the speed limit

.7 \* .9 = 63% of drivers going 20 MPH or more over the speed limit are pulled over

.2 \* .05 = 1% of drivers going less than 20 MPH over the speed limit are pulled over

==> This means 64% who are speeding are pulled over

P(PO | Speeding) = (P(Speeding | PO) (P(PO)) / P (Speeding)

P(Speeding | PO) = P(GT20 U LT20 | PO)

= P(GT20 | PO) + P(LT20 | PO)

= .63 + .01

= .64

.3 \* .9 = 27% are speeding by 20 MPH or more and not being pulled over

.8 \* .2 = 16% are speeding by less than 20 MPH and not being pulled over

==> This means 53% who are speeding are not pulled over

P(PO | Speeding) = (P(Speeding | PO) \* P (PO)) / P(Speeding)

P(Speeding) = P(Speeding, PO) + P(Speeding, ~PO)

= (.64 \* .95) + (.53 \* .36)

=. 608 + .191

=.799

P(PO | Speeding) = (P(Speeding | PO) \* P (PO)) / P(Speeding)

= (.64 \* .95) / ( (.64 \* .95) + (.53 \* .36))

= .608 / .799

= .761

1. (6 points) Machine learning
   1. What is the difference between supervised and unsupervised learning?

Supervised learning - agent is given set of examples (i.e., training set) and told which are members of category and which are not. Based on this, the agent learns to classify new examples it has been given

Unsupervised learning - agent is only given input without any feedback. For example, the agent may make recommendations for subsequent purchases based on monitoring a person’s shopping habits

* 1. List at least 3 metrics which are used to evaluate supervised learning algorithms.

Precision

Recall

F Measure

tp = true positive

fp = false positive

fn = false negative

tn = true negative

Precision = tp / (tp + fp)

Recall = tp / (tp + fn)

F measure = (2 \* Recall \* Precision) / (Recall + Precision)

* 1. Explain what cross-validation is.

Cross-validation is a technique to evaluate ML models by training them on subsets of input data. The data is split into k subsets (called folds) and the model is trained on (k-1) of the subsets and tested on the subset that is not used for training. This is repeated k times, using a different subset for testing each time. For example, for 4-fold cross validation, the data is split into 4 subsets and 3 of these are used for training and one is used for testing. This is repeated 4 times, using a different subset for testing each time.