CSCI 5582
Artificial Intelligence

Lecture 28

Jim Martin

HW 3

- On the first set the average accuracy was 87% with 11 submissions at 100%.
- On the second set the average accuracy was 76% with 2 submissions getting 100%
  - One of those was a rule-based approach
    - With basically 1 simple rule and a variant on it.

Final Details

- Monday, 1:30PM, here. It will be 2 ½ hours.
  - Come on time, spread out, bring a calculator, don't bring the rest of all your worldly belongings, probably ought to use a pencil and eraser.

Today 12/14

- Final Review

Final Topics

- Search
- Representation
- Uncertainty
- Machine Learning
- Language Processing

Meta Topics

- There are connections among all the topics. Search, representation, probability and learning are all intertwined.
- I may ask questions that make you make connections

Final

- Each section will have a similar structure to the quizzes. Easy factual stuff, followed by a couple of problems to work out that demonstrate understanding.

Final

- Material that I asked you to prepare for but was not covered on a quiz is fair game.

Final

- General Hints:
  - I will never ask a question that requires you to transcribe an algorithm. If you find yourself doing that you should stop and re-read the question.
  - You do however need to know (understand, grok, grasp) the algorithms to answer questions about them.

Final Hints: Example

- What kind of search is the DT learning algorithm?
- Is it optimal? Why?
- Is Neural Net learning a search?
- How does the choice of k in k-dl lists effect the likelihood of the DL learning algorithm finding a reasonable list.

Final Hints

- Some of you should really consider pencil (and an eraser).
- You should bring a calculator if it makes you feel better
  - Arithmetic errors that arise in computing the right thing won't hurt you (much)
  - Exact answers to the wrong thing will

Search

- State-space search
- Optimization/iterative improvement
- Constraint-based search

State-Space Search

- Basic algorithms
- A*
- IDA*

- How they relate to each other

Optimization

- Annealing, hill-climbing, random restart hill-climbing.
- The nature of the states, the problems you run into and how annealing or random-restart address the problems.

Constraint-Based Searches

- What's a constraint? What's a problem?
- Backtracking methods
- Min-conflict/satisfiability methods
- What's the connection between satisfiability and propositional logic?

Representation and Reasoning

- Propositional logic and reasoning with it.
- First order logic and reasoning with it.

Propositional Logic

- Syntax and Semantics
- Proving stuff
- Wumpus world

First Order Logic

- Focus here will be on representing stuff of interest rather than on proving things.
- Although that doesn't mean I won't give a simple backward or forward chaining example

Representation and Reasoning Hints

- If I say use Propositional Logic, use Propositional Logic.
- If I ask what does the agent know at some point in time, show me the strongest thing you can say.
- If I give a problem to solve using logic, then I want you to show how a machine could solve it mechanically. Not that you as a human can figure it out.

Hints

- That technique you can't remember the name of is called *Resolution*.
- You can't just randomly re-order ands and ors until you get something you like.

Example

- You know
  1. A
  2. A->B^C
  3. C->D

  Prove D

- MP with 1&2 produces (4) B^C
- AE on 4 produces

(5) B and (6) C

- MP with 3&6 produces D. Done.

Wumpus World

- Or something like it.
  - Rules are either given or you know them
  - B11 -> Pit1,2 or Pit2,1 etc
- Agent moves from here to there, and detects this and that, what do you know.

Uncertainty

- Basic probability material
- Bayesian reasoning
- Bayesian belief nets
- Hidden Markov models
- Naïve Bayes classification
  - How they all connect

Basic Material

- Basic syntax, semantics and definitions.
- Memorize the definition of a conditional probability
  - P(A|B) = P(A^B)/P(B)

Basic Material

- Argmax P(X|Y) where choosing X means choosing the right X from some set of choices (diseases, classes, tags, words, whatever)
- Using Bayes when the data for P(X|Y) can't be gotten.

Basic Material

- For Bayesian diagnosis questions, there's a query about some state of affairs and there's evidence…

$$P(State|E) = P(E|State)P(State)/P(E)$$

Bayesian Belief Nets

- Syntax and semantics
- It's a way of encoding the joint probability distribution of the variables in the network.
- The entries are based on the shape of the network.
- The network can only directly answer questions concerning the conjunctive status of all the variables in the network

BBN Examples

1. Think about what the question is asking: is there evidence or not?
2. Formulate the question as a probability to be assessed.
3. Ask yourself if this is the kind of probability that the belief net can answer directly or is it something that requires multiple queries?

BBN Examples

- For example, I give you some evidence e, and ask you about a variable q, given some network.
  - That's P(q|e) with the network in the background
  - The belief net can't answer that directly
  - But you can re-write it as a ratio
    - P(q^e)/P(e)

BBN Examples

- But it probably can't answer that either.
  - It can answer questions about conjunctive states of ALL the variables.
    - P(q^e^ configurations of the remaining vars)
    - Same for P(e)
    - You sum the non-overlapping configurations.

Belief Revision

- There is often a question that goes like this:
  - Here's a fact. What should you believe about variable X now.
  - Here's another fact. Now what do you believe about X
- These questions are cumulative. You know the first fact, and then the first fact AND the second fact.

Hint

- We talked about basics of probability, diagnosis (stiff necks), naïve Bayes, Markov assumptions, and then belief nets
- They're all related… belief nets capture conditional independence assumptions; naïve Bayes and Markov models are based on independence assumptions.

Machine Learning

- Mainly on supervised machine learning
  - Organization of training
  - Kinds of learning and things learned
    - Trees, lists, etc
  - Meta-issues: where does the hypothesis space come from, what effect does the size of the space have on learning?
  - Boosting

Decision Trees

- Definitions of trees
- How they work
- How they're learned

Choosing an Attribute

- Approximation to the Information Gain metric.
  - Figure out your original error rate
  - Apply a feature which branches N ways
  - Divide the training data along the branches
  - Count the labels at each leaf and pick the majority label
  - How many do you get right?

Note

- This technique indirectly gets at the notion of trying to find small trees with uniform leaves.

Note

- The entire training set is available only at the top of the tree.
- Once a feature has been placed into the tree, the training data splits according to the values of the feature. Ie. Choosing tests deeper in the tree involves a subset of the original set.

Decision Lists

- Search for sequences of tests that cover subsets of the training data.
- An instance that passes a test is assigned a label
- An instance that doesn't pass a test is passed to the next test.

Decision Lists

- Its useful to talk about
  - Accuracy of a test (how well does it predict the right answer for the instances it covers)
  - Coverage of a test (how many instances does it apply to?)
  - The book's algorithm is looking for tests of length k with 100% accuracy
  - All things being equal we like tests with higher coverage (why?)

Why?

- Occam's razor
  - Prefer simple hypotheses to complex ones
  - Choosing tests with large coverage reduces the examples passed on to the rest of the algorithm
    - Leading it to terminate sooner
      - Leading to smaller lists
        - » Making Occam happy

Decision Lists

- The boxes and arrows seemed to confuse folks. Its really just an ordered list of tests
  - Test -> label
  - Test -> label
  - Test -> label
  - …
  - Emit the label attached to the first test that succeeds and then stop

Hints

- For DLs two-label (binary) tasks lend themselves to techniques that don't really generalize
  - I.e. If I start with 5 yesses and 5 nos, and I can knock out 4 yesses with the first test
  - Then I might choose to worry about catching that last yes, rather than covering a larger number of the nos

Ensembles

- Know the basic idea of how ensembles work.
  - Some way of producing independent classifiers
  - A voting scheme

Other Classifiers

- SVMs, Neural Nets
  - Just need a superficial familiarity with the basic ideas.

Language Processing

- Mainly the connections to other topics in the course
  - How can language problems be viewed as probability problems?
  - Machine learning problems?

Language Processing

- I won't ask a specific detailed MT question…
- Think of generative probabilistic sequence applications that are language related
  - Speech (audio to text)
  - MT (German to English)
  - OCR (pixels to texts)
  - IE (texts to database entries)

Generative Statistical Models

- Underlying (hidden) states in a statistical machine…
- Hidden states emit outputs (observables)
- Want to infer the hidden processing from the observables
  - In other words the observables are what you have, the hidden states are what you want.