

Final examination information

The final examination is on Tuesday December 16 from 6:30pm until 9:30pm in DCC 318. You may feel free to bring food as long as you clean up after yourself. The examination is closed book and closed notes. No calculators are allowed or necessary; you will probably have to do a little simple arithmetic.

The examination will be designed to test both:

- conceptual understanding — the ideas behind the algorithms, which algorithm to apply to a problem and what the tradeoffs are
- detailed understanding — the intricacies of how an algorithm works and issues in its implementation.

There will be some questions involving factual recall or simple explanation of concepts or algorithms, but there will also be questions asking you to apply the course material to various problems and situations. You may be asked to extrapolate from course material or apply related concepts to new problems; I think one characteristic of a good examination is that students should learn something from it.

I will release on the web page the midterm examinations for this class from Fall 1999 and 2000. I will not be releasing previous years' final examinations. I highly recommend reviewing your quizzes.

Formulas provided on the final examination

- Bayes classifiers (see slides or online reference)
- Sequential decision problems & reinforcement learning:

$$v = \operatorname{argmax}_{v_j \in V} \sum_{h_i \in H} P(v_j|h_i)P(h_i|D) \quad U(s) = R(s) + \gamma \max_a \sum_{s'} T(s, a, s')U(s') \quad (17.5)$$

$$v = \operatorname{argmax}_{v_j \in V} P(v_j) \prod_i P(a_i|v_j) \quad U(s) = R(s) + \gamma \sum_{s'} T(s, \pi(s), s')U(s') \quad (17.10)$$

$$h = \operatorname{argmax}_{h_i \in H} P(D|h_i)P(h_i) \quad U_{i+1}(s) \leftarrow R(s) + \gamma \max_a \sum_{s'} T(s, \pi(s), s')U_i(s') \quad (17.6)$$

- Perceptron learning

$$W_j \leftarrow W_j + \alpha \times \text{Err} \times g'(in) \times x_j \quad (20.12) \quad U^\pi(s) \leftarrow U^\pi(s) + \alpha(R(s) + \gamma U^\pi(s') - U^\pi(s)) \quad (21.3)$$

$$Q(a, s) \leftarrow Q(a, s) + \alpha(R(s) + \gamma \max_{a'} Q(a', s') - Q(a, s)) \quad (21.8)$$

- Backpropagation (see handout)

$$\vec{W}_j^{i,n} \leftarrow \vec{W}_j^{i,n} + \alpha \times \vec{a}_j \times \Delta_{i,n}$$

$$\vec{W}_k^{j,m} \leftarrow \vec{W}_k^{j,m} + \alpha \times \vec{a}_k \times \Delta_{j,m}$$

$$\vec{W} \leftarrow \vec{W} + \alpha \times \vec{I} \times \text{Err}$$

$$\Delta_{i,n} = \text{Err}_{i,n} \times g'(in_{i,n})$$

$$\Delta_{j,m} = g'(in_{j,m}) \sum_{n=1}^r W_{j,m}^{i,n} \times \Delta_{i,n}$$

$$g(x) = \frac{1}{1 + e^{-x}}$$

- Information

$$I(P(v_1), \dots, P(v_n)) = \sum_i -P(v_i) \log_2 P(v_i) \quad (\text{p. 659})$$

Final examination topics

| | |
|---|-------|
| Introduction | |
| What is AI? | 1.1 |
| Agent structure & environments | 2 |
| Search | |
| Blind search | |
| Formulating search problems | 3.1–2 |
| State space versus search tree | 3.3 |
| Optimality, completeness, time & space complexity | 3.3 |
| Six blind searches | 3.4 |
| Avoiding repeated states | 3.5 |
| Heuristic search | |
| Greedy search | 4.1 |
| A* search search | 4.1 |
| Heuristic functions | 4.2 |
| Admissibility, monotonicity / consistency | 4.1–2 |
| Memory bounded A* algorithms | 4.1 |
| Iterative improvement algorithms | |
| Hill climbing | 4.3 |
| Simulated annealing | |
| Local beam search | |
| Genetic Algorithms | |
| Constraint satisfaction problems | |
| CSP & assignment problems | 5.1 |
| Constructive approaches | 5.2 |
| Blind search approaches | |
| Backtracking, forward checking | |
| Heuristics to improve blind search strategies | |
| Constraint propagation | |
| Repair approaches | 5.3 |
| Min-conflicts heuristic | |
| Game playing search | |
| MINIMAX search | 6.1–2 |
| Perfect vs. imperfect decisions | 6.4 |
| Evaluation functions | 6.4 |
| Alpha-beta pruning | 6.3 |
| Probabalistic games (EXPECTIMAX) | 6.5 |
| Logic | |
| Knowledge representation & logical systems | |
| Inference & entailment | 7.1–3 |
| Soundness & completeness | |
| Equivalence | 7.5 |
| Propositional logic | |
| Horn normal form | 7.4 |
| Conjunctive & implicative normal forms | 7.5 |
| Inference in propositional logic | |
| Forward and backward chaining | 7.5 |
| Resolution refutation proofs with set of support strategy | |

| | |
|---|----------------------------|
| Logic, continued | |
| First order logic | 7.1–3 |
| Quantifiers, Inference in first order logic | 9.1 |
| Horn normal form | |
| Conjunctive normal form 9.5 | |
| Skolemization | |
| Unification | 9.2 |
| Forward and backward chaining | 9.3–4 |
| Resolution refutation proofs with set of support strategy | 9.5 |
| Gödel’s completeness and incompleteness theorems | p. 295, p. 302 |
| Learning | |
| Introduction | 18.1–2 |
| Classification problems | |
| Decision trees | 18.3–4 |
| Overfitting | |
| χ^2 pruning | |
| Gain ratio | slides |
| Rule post pruning | slides |
| Neural networks | 20.5 |
| Perceptrons | |
| Perceptron learning rule | |
| Representational power of perceptrons | |
| Multilayer feed-forward networks | |
| Sigmoid units | |
| Backpropagation | 20.5 & handout |
| Representational power | |
| Bayesian learning/classifiers | slides, reserves, (20.1–2) |
| Probability basics | 13 |
| Conditional independence | |
| Bayes rule | |
| Brute force classifier | |
| Optimal classifier | |
| Naive classifier | |
| m-estimates | |
| Reinforcement learning | |
| Introduction | 21.1 |
| Utility | 16.1–3 |
| Sequential decision problems | 17.1–3 |
| Value iteration | |
| Policy iteration | |
| Passive reinforcement learning | 21.2 |
| Direct utility estimation | |
| Adaptive dynamic programming | |
| Temporal differencing | |
| Active reinforcement learning | 21.3 |
| Exploration | |
| Q-learning | |

Key algorithms/techniques

These are most likely to be the subject of a problem in which you are asked to solve a specific problem using a given algorithm/technique.

Blind search

Breadth first search
Depth first search
Depth-limited search
Iterative deepening search
Uniform cost search
Bi-directional search

Heuristic search

Greedy search
A* search

Constraint satisfaction search

Constructive methods (with forward checking, constraint propagation, and heuristics)
Heuristic repair (with min-conflicts heuristic)

Game playing search

MINIMAX
MINIMAX with alpha-beta pruning

Logic

Translating/transforming into logic sentences and into normal forms
Forward chaining
Backward chaining
Resolution refutation proof with set of support strategy

Decision trees

Decision tree learning with information gain heuristic
Decision tree learning with χ^2 pruning and with gain ratio
Rule post-pruning

Artificial neural networks

Perceptron learning rule
Backpropagation

Bayes classifiers

Bayes naive classifier

Reinforcement learning

Value iteration
Policy iteration
Adaptive dynamic programming
Temporal differencing
Q-learning