Course Subject, Number and Title
Computer Sciences 540: Introduction to Artificial Intelligence

Credits: 3

Course Designations and Attributes
Breadth - Natural Science, Level – Advanced, L&S Credit - Counts as Liberal Arts and Science credit in L&S

Meeting Time and Location
Tuesdays and Thursdays 9:30 – 10:45 a.m. in room 125 Agricultural Hall

Instructional Mode: Lecture, all face-to-face

Specify how Credit Hours are met by the Course
Traditional Carnegie Definition – Two 75-minute faculty-taught classroom lectures and a minimum of two hours of out-of-class student work each week over approximately 15 weeks

INSTRUCTOR

Instructor: Professor Charles (Chuck) Dyer

Instructor Office Hours: Tuesdays and Thursdays 2:30 – 4:00 p.m.

Instructor Email: dyer@cs.wisc.edu

OFFICIAL COURSE DESCRIPTION

Course Description
Principles of knowledge-based search techniques, automatic deduction, knowledge representation using predicate logic, machine learning, probabilistic reasoning. Applications in tasks such as problem solving, data mining, game playing, natural language understanding, computer vision, speech recognition, and robotics.

Requisites
(COMP SCI 300 or 367) and (MATH 211, 217, 221, or 275) or graduate or professional standing or declared in the Capstone Certificate in Computer Sciences for Professionals

LEARNING OUTCOMES

Course Learning Outcomes

Students are expected to know the following upon completion of this course:

Uninformed Search Methods – Be able to formulate problem solving tasks as searching a state space graph, problem representation in terms of states, goal test, operators, state-space graph search formulation, closed world assumption, expanding a node, frontier list, partial solution path, solution path, search tree, breadth-first search, depth-first search, chronological backtracking, uniform-cost search, iterative-deepening search, bidirectional search, completeness, optimality, admissibility, time and space complexity, detecting repeated states, explored list.

Informed Search Methods – Understand heuristic functions, evaluation functions, best-first search, greedy best-first search, beam search, algorithm A, algorithm A*, admissible heuristic, consistent heuristic, better informed heuristic, devising heuristics.

Local Search Methods – Local search problem formulation, operators, neighborhood, move set, hill-climbing algorithm, local optima problem, hill-climbing with random restarts, stochastic hill-climbing (simulated annealing) algorithm, escaping local optima, Boltzman's equation, cooling schedule, genetic algorithms, crossover, mutation, fitness function, proportional fitness selection, population, crowding.

Game Playing – Zero-sum games, perfect information games, deterministic vs. stochastic games, game playing as search, search tree, branching factor, ply, minimax principle, minimax algorithm, static evaluation function, alpha-beta pruning, cutoff, alpha-beta pruning algorithm, best case and worst case of alpha-beta vs. minimax, iterative-deepening with alpha-beta, horizon effect, quiescence search, representing non-deterministic games, chance nodes, expectimax value, Monte Carlo tree search.

Constraint Satisfaction - Problem formulation in terms of variables, domains and constraints, constraint graph, depth-first search, backtracking with consistency checking, most constrained variable heuristic, most constraining variable heuristic, least constraining value heuristic, min-conflicts heuristic, min-conflicts algorithm, forward checking algorithm, arc consistency algorithm (AC-3).

Unsupervised Learning – Inductive learning problem, unsupervised learning problem, feature space, feature, attribute, examples, labels, classes, training set, testing set, classification problems, inductive bias, preference bias, hierarchical agglomerative clustering algorithm, single linkage, complete linkage, average linkage, dendrogram, *k*-means clustering algorithm, cluster center, distortion cluster quality.

***K*-nearest Neighbors and Decision Trees** – *K*-nearest neighbor algorithm, Ockham's razor, decision tree algorithm, information gain, max-gain, entropy, conditional entropy, remainder, overfitting problem, pruning, training set, testing set, tuning set, setting parameters, *k*-fold cross validation, leave-one-out cross validation, random forests, bagging ensemble learning.

Support Vector Machines – Maximum margin, definition of margin, kernel trick, support vectors, slack variables.

Neural Networks – Perceptron,LTU, activation functions, bias input, input units, output units, Perceptron learning rule, Perceptron learning algorithm, epoch, weight space, input space, linearly separable, credit assignment problem, multi-layer feed-forward networks, hidden units, sigmoid function, ReLU, back-propagation algorithm, gradient descent search in weight space, deep learning, convolutional neural networks, pooling.

Reasoning under Uncertainty – Random variable, mutually exclusive, prior probability, 3 axioms of probability, joint probability, conditional probability, posterior probability, full joint probability distribution, degrees of freedom, summing out, marginalization, normalization, product rule, chain rule, conditionalized version of chain rule, Bayes's rule, conditionalized version of Bayes's rule, addition/conditioning rule, independence, conditional independence, naïve Bayes classifier.

Bayesian Networks – Bayesian network DAG, conditional probability tables, space saving compared to full joint probability distribution table, conditional independence property defined by a Bayesian network, inference by enumeration from a Bayesian network, naïve Bayes classifier as a Bayesian network.

Speech Recognition – Phones, phonemes, speech recognition using Bayes's rule, language model, acoustic model, bigram model, trigram model, first-order Markov assumption, probabilistic finite state machine, first-order Markov model, state transition matrix, π vector, computing conditional probabilities from a Markov model, hidden Markov model, observation likelihood matrix, computing joint probabilities and conditional probabilities from an HMM by enumeration.

Computer Vision – Viola-Jones face detection algorithm, boosting ensemble learning, AdaBoost algorithm, Eigenfaces algorithm, nearest-neighbor classification, image space, face space, average face, eigenvalues, eigenvectors, dimensionality reduction.

GRADING

- Homeworks: 40% (5 assignments, each worth 8%)
- Midterm exam: 30%
- Final exam: 30%

Attendance and class participation are *not* used as part of the grading. Final grades are curved based on the scores of all the undergraduates in the course, with approximately 20% A's, 15% AB's, 23% B's, 15% BC's, 20% C's, 5% D's, 2% F's

DISCUSSION SESSIONS

There are no discussion sessions for this course but there are peer mentors who are available to answer course content questions and questions on the homework.

LABORATORY SESSIONS

There are no laboratory sessions for this course. Students may do their programming assignments using either their own computer or else one of the computers in the Computer Sciences Department's instructional labs.

REQUIRED TEXTBOOK, SOFTWARE & OTHER COURSE MATERIALS

Required text: S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, 3rd ed., Pearson, 2010. Other readings will be made available electronically on the course web page. Programming assignments will be done using the Java programming language.

EXAMS

The midterm exam will cover material in the first half of the course. It will be taken during a 2-hour block in an evening. Students may bring one 8.5” x 11” sheet of notes on both sides, but otherwise the exam is closed-book. A calculator may also be used, but not one on a phone. Make-up exams must be approved at least one week before the regular exam.

The final exam will cover material after the midterm exam, so it will *not* be cumulative. It will be taken during a 2-hour block as assigned by the university. Students may bring one 8.5” x 11” sheet of notes on both sides, but otherwise the exam is closed-book. A calculator may also be used, but not one on a phone. No make-up final exam is possible except as allowed by university policy.

HOMEWORK & OTHER ASSIGNMENTS

Homework assignments will consist of written problems and programming problems. Programming problems will require writing code in the Java programming language. Some supplied skeleton Java code may also be given as a starting point. All homework is to be completed individually. Students may do their programming assignments using either their own computer or else one of the computers in the Computer Sciences Department’s instructional labs. Answers to written problems and Java code that is written by the student will be handed in electronically using the UW-Madison’s Canvas system.

OTHER COURSE INFORMATION

None

RULES, RIGHTS & RESPONSIBILITIES

See the UW-Madison’s Guide to Rules, Rights and Responsibilities

ACADEMIC INTEGRITY

By enrolling in this course, each student assumes the responsibilities of an active participant in UW-Madison’s community of scholars in which everyone’s academic work and behavior are held to the highest academic integrity standards. Academic misconduct compromises the integrity of the university. Cheating, fabrication, plagiarism, unauthorized collaboration, and helping others commit these acts are examples of academic misconduct, which can result in disciplinary action. This includes but is not limited to failure on the assignment/course, disciplinary probation, or suspension. Substantial or repeated cases of misconduct will be forwarded to the Office of Student Conduct & Community Standards for additional review. For more information, refer to studentconduct.wiscweb.wisc.edu/academic-integrity/.

ACCOMMODATIONS FOR STUDENTS WITH DISABILITIES

McBurney Disability Resource Center syllabus statement: “The University of Wisconsin-Madison supports the right of all enrolled students to a full and equal educational opportunity.

The Americans with Disabilities Act (ADA), Wisconsin State Statute (36.12), and UW-Madison policy (Faculty Document 1071) require that students with disabilities be reasonably accommodated in instruction and campus life. Reasonable accommodations for students with disabilities is a shared faculty and student responsibility. Students are expected to inform faculty of their need for instructional accommodations by the end of the third week of the semester, or as soon as possible after a disability has been incurred or recognized. Faculty will work either directly with the student or in coordination with the McBurney Center to identify and provide reasonable instructional accommodations. Disability information, including instructional accommodations as part of a student's educational record, is confidential and protected under FERPA." http://mcburney.wisc.edu/facstaffother/faculty/syllabus.php

DIVERSITY & INCLUSION

Institutional statement on diversity: "Diversity is a source of strength, creativity, and innovation for UW-Madison. We value the contributions of each person and respect the profound ways their identity, culture, background, experience, status, abilities, and opinion enrich the university community. We commit ourselves to the pursuit of excellence in teaching, research, outreach, and diversity as inextricably linked goals.

The University of Wisconsin-Madison fulfills its public mission by creating a welcoming and inclusive community for people from every background – people who as students, faculty, and staff serve Wisconsin and the world." https://diversity.wisc.edu/