Artificial Intelligence (AI)

Preamble

Artificial intelligence (AI) studies problems that are difficult or impractical to solve with traditional algorithmic approaches. These problems are often reminiscent of those considered to require human intelligence, and the resulting AI solution strategies typically generalize over classes of problems. AI techniques are now pervasive in computing, supporting everyday applications such as email, social media, photography, financial markets, and intelligent virtual assistants (e.g., Siri, Alexa). These techniques are also used in the design and analysis of autonomous agents that perceive their environment and interact rationally with it, such as self-driving vehicles and other robots.

Traditionally, AI has included a mix of symbolic and subsymbolic approaches. The solutions it provides rely on a broad set of general and specialized knowledge representation schemes, problem solving mechanisms, and optimization techniques. These approaches deal with perception (e.g., speech recognition, natural language understanding, computer vision), problem solving (e.g., search, planning, optimization), generation (e.g., narrative, conversation, images, models, recommendations), acting (e.g., robotics, task-automation, control), and the architectures needed to support them (e.g., single agents, multi-agent systems). Machine learning may be used within each of these aspects and can even be employed end-to-end across all of them. The study of Artificial Intelligence prepares students to determine when an AI approach is appropriate for a given problem, identify appropriate representations and reasoning mechanisms, implement them, and evaluate them with respect to both performance and their broader societal impact.

Over the past decade, the term “artificial intelligence” has become commonplace within businesses, news articles, and everyday conversation, driven largely by a series of high-impact machine learning applications. These advances were made possible by the widespread availability of large datasets, increased computational power, and algorithmic improvements. In particular, there has been a shift from engineered representations to representations learned automatically through optimization over large datasets. The resulting advances have put such terms as “neural networks” and “deep learning” into everyday vernacular. Businesses now advertise AI-based solutions as value-additions to their services, so that “artificial intelligence” is now both a technical term and a marketing buzzword. Other disciplines, such as biology, art, architecture, and finance, increasingly use AI techniques to solve problems within their disciplines.

For the first time in our history, the broader population has access to sophisticated AI-driven tools, including tools to generate essays or poems from a prompt, artwork from a description, and fake photographs or videos depicting real people. AI technology is now in widespread use in stock trading, curating our news and social media feeds, automated evaluation of job applicants, detection of medical conditions, and influencing prison sentencing through recidivism prediction. Consequently, AI technology can have significant societal impacts and ethical considerations that must be understood and considered when developing and applying it.

Changes since CS2013

To reflect this recent growth and societal impact, the knowledge area has been revised from CS2013 in the following ways.

- The name has changed from "Intelligent Systems" to "Artificial Intelligence," to reflect the most common terminology used for these topics within the field and its more widespread use outside the field.
- An increased emphasis on neural networks and representation learning reflects the recent advances in the field. Given its key role throughout AI, search is still emphasized but there is a slight reduction on symbolic methods in favor of understanding subsymbolic methods and learned representations. It is important, however, to retain knowledge-based and symbolic approaches within the AI curriculum because these methods offer unique capabilities, are used in practice, ensure a broad education, and because more recent neurosymbolic approaches integrate both learned and symbolic representations.
- There is an increased emphasis on practical applications of AI, including a variety of areas (e.g., medicine, sustainability, social media). This includes explicit discussion of tools that employ deep generative models (e.g., ChatGPT, DALL-E, Midjourney) and are now in widespread use, covering how they work at a high level, their uses, and their shortcomings/pitfalls.
- The curriculum reflects the importance of understanding and assessing the broader societal impacts and implications of AI methods and applications, including issues in AI ethics, fairness, trust, and explainability.
- The AI knowledge area includes connections to data science through 1) cross-connections with the Data Management and other knowledge areas and 2) a sample Data Science model course.
- There are explicit goals to develop basic AI literacy and critical thinking in every computer science student, given the breadth of interconnections between AI and other knowledge areas in practice.

Consider recent AI advances when using this curriculum

The field of AI is undergoing rapid development and increasingly widespread applications. Since the first draft of this document, several new techniques (e.g., generative networks, large language models) have become widely used and so were added to the CS or KA Cores. This document is as current as we can make it in 2023. However, we expect such rapid changes to continue in the subfield of AI during the expected life of this document. Consequently, it is imperative that faculty teaching AI understand current advances and consider whether these advances should be taught in order to keep the curriculum current.

Core Hours

Knowledge Unit	**CS Core**	**KA Core**
Fundamental Issues	2	1
Search	2 + 3 (AL)	6

Fundamental Knowledge Representation and Reasoning	1 + 1 (MSF)	2
Machine Learning	4	6
Applications and Societal Impact	3	3
Probabilistic Representation and Reasoning		
Planning		
Logical Representation and Reasoning		
Agents and Cognitive Systems		
Natural Language Processing		
Robotics		
Perception and Computer Vision		
Total	**12**	**18**

The CS Core includes 3 hours that are shared with and counted under Algorithm Foundations (Uninformed search) and 1 hour that is shared with and counted under Mathematical Foundations (Probability).

Knowledge Units

AI-Introduction: Fundamental Issues

CS Core:

1. Overview of AI problems, Examples of successful recent AI applications
2. Definitions of agents with examples (e.g., reactive, deliberative)
3. What is intelligent behavior?
   a. The Turing test and its flaws
   b. Multimodal input and output
   c. Simulation of intelligent behavior
   d. Rational versus non-rational reasoning
4. Problem characteristics
   a. Fully versus partially observable
   b. Single versus multi-agent
   c. Deterministic versus stochastic
   d. Static versus dynamic
   e. Discrete versus continuous
5. Nature of agents

  a. Autonomous, semi-autonomous, mixed-initiative autonomy
  b. Reflexive, goal-based, and utility-based
  c. Decision making under uncertainty and with incomplete information
  d. The importance of perception and environmental interactions
  e. Learning-based agents
  f. Embodied agents
     i. sensors, dynamics, effectors
6. Overview of AI Applications, growth, and impact (economic, societal, ethics)

KA Core:

7. Practice identifying problem characteristics in example environments
8. Additional depth on nature of agents with examples
9. Additional depth on AI Applications, Growth, and Impact (economic, societal, ethics, security)

Non-core:

10. Philosophical issues
11. History of AI

Illustrative Learning Outcomes:

1. Describe the Turing test and the “Chinese Room” thought experiment.
2. Differentiate between optimal reasoning/behavior and human-like reasoning/behavior.
3. Differentiate the terms: AI, machine learning, and deep learning.
4. Enumerate the characteristics of a specific problem.

AI-Search: Search

CS Core:

1. State space representation of a problem
   a. Specifying states, goals, and operators
   b. Factoring states into representations (hypothesis spaces)
   c. Problem solving by graph search
      i. e.g., Graphs as a space, and tree traversals as exploration of that space
      ii. Dynamic construction of the graph (not given upfront)
2. Uninformed graph search for problem solving (See also: AL-Foundational)
   a. Breadth-first search
   b. Depth-first search
      i. With iterative deepening
   c. Uniform cost search
3. Heuristic graph search for problem solving (See also: AL-Strategies)
   a. Heuristic construction and admissibility
   b. Hill-climbing
   c. Local minima and the search landscape
      i. Local vs global solutions
   d. Greedy best-first search
   e. A* search

4. Space and time complexities of graph search algorithms

KA Core:

5. Bidirectional search
6. Beam search
7. Two-player adversarial games
    a. Minimax search
    b. Alpha-beta pruning
        i. Ply cutoff
8. Implementation of A* search
9. Constraint satisfaction

Non-core:

10. Understanding the search space
    a. Constructing search trees
    b. Dynamic search spaces
    c. Combinatorial explosion of search space
    d. Search space topology (e.g., ridges, saddle points, local minima)
11. Local search
12. Tabu search
13. Variations on A* (IDA*, SMA*, RBFS)
14. Two-player adversarial games
    a. The horizon effect
    b. Opening playbooks/endgame solutions
    c. What it means to “solve” a game (e.g., checkers)
15. Implementation of minimax search, beam search
16. Expectimax search (MDP-solving) and chance nodes
17. Stochastic search
    a. Simulated annealing
    b. Genetic algorithms
    c. Monte-Carlo tree search

Illustrative Learning Outcomes:

1. Design the state space representation for a puzzle (e.g., N-queens or 3-jug problem)
2. Select and implement an appropriate uninformed search algorithm for a problem (e.g., tic-tac-toe), and characterize its time and space complexities.
3. Select and implement an appropriate informed search algorithm for a problem after designing a helpful heuristic function (e.g., a robot navigating a 2D gridworld).
4. Evaluate whether a heuristic for a given problem is admissible/can guarantee an optimal solution.
5. Apply minimax search in a two-player adversarial game (e.g., connect four), using heuristic evaluation at a particular depth to compute the scores to back up. [KA Core]
6. Design and implement a genetic algorithm solution to a problem.
7. Design and implement a simulated annealing schedule to avoid local minima in a problem.

8. Design and implement A*/beam search to solve a problem, and compare it against other search algorithms in terms of the solution cost, number of nodes expanded, etc.
9. Apply minimax search with alpha-beta pruning to prune search space in a two-player adversarial game (e.g., connect four).
10. Compare and contrast genetic algorithms with classic search techniques, explaining when it is most appropriate to use a genetic algorithm to learn a model versus other forms of optimization (e.g., gradient descent).
11. Compare and contrast various heuristic searches vis-a-vis applicability to a given problem.
12. Model a logic or Sudoku puzzle as a constraint satisfaction problem, solve it with backtrack search, and determine how much arc consistency can reduce the search space.

AI-KRR: Fundamental Knowledge Representation and Reasoning

CS Core:

1. Types of representations
    a. Symbolic, logical
        i. Creating a representation from a natural language problem statement
    b. Learned subsymbolic representations
    c. Graphical models (e.g., naive Bayes, Bayesian network)
2. Review of probabilistic reasoning, Bayes theorem (See also: MSF-Probability)
3. Bayesian reasoning
    a. Bayesian inference

KA Core:

4. Random variables and probability distributions
    a. Axioms of probability
    b. Probabilistic inference
    c. Bayes' Rule (derivation)
    d. Bayesian inference (more complex examples)
5. Independence
6. Conditional Independence
7. Markov chains and Markov models
8. Utility and decision making

Illustrative Learning Outcomes:

1. Given a natural language problem statement, encode it as a symbolic or logical representation.
2. Explain how we can make decisions under uncertainty, using concepts such as Bayes theorem and utility.
3. Compute a probabilistic inference in a real-world problem using Bayes' theorem to determine the probability of a hypothesis given evidence.
4. Apply Bayes' rule to determine the probability of a hypothesis given evidence.
5. Compute the probability of outcomes and test whether outcomes are independent.

AI-ML: Machine Learning

CS Core:

1. Definition and examples of a broad variety of machine learning tasks
   a. Supervised learning
      i. Classification
      ii. Regression
   b. Reinforcement learning
   c. Unsupervised learning
      i. Clustering
2. Fundamental ideas:
   a. No free lunch theorem: no one learner can solve all problems; representational design decisions have consequences.
   b. Sources of error and undecidability in machine learning
3. A simple statistical-based supervised learning such as linear regression or decision trees
   a. Focus on how they work without going into mathematical or optimization details; enough to understand and use existing implementations correctly
4. The overfitting problem/controlling solution complexity (regularization, pruning – intuition only)
   a. The bias (underfitting) – variance (overfitting) tradeoff
5. Working with Data
   a. Data preprocessing
      i. Importance and pitfalls of preprocessing choices
   b. Handling missing values (imputing, flag-as-missing)
      i. Implications of imputing vs flag-as-missing
   c. Encoding categorical variables, encoding real-valued data
   d. Normalization/standardization
   e. Emphasis on real data, not textbook examples
6. Representations
   a. Hypothesis spaces and complexity
   b. Simple basis feature expansion, such as squaring univariate features
   c. Learned feature representations
7. Machine learning evaluation
   a. Separation of train, validation, and test sets
   b. Performance metrics for classifiers
   c. Estimation of test performance on held-out data
   d. Tuning the parameters of a machine learning model with a validation set
   e. Importance of understanding what a model is doing, where its pitfalls/shortcomings are, and the implications of its decisions
8. Basic neural networks
   a. Fundamentals of understanding how neural networks work and their training process, without details of the calculations
   b. Basic introduction to generative neural networks (e.g., large language models)
9. Ethics for Machine Learning (See also: SEP-Context)
   a. Focus on real data, real scenarios, and case studies
   b. Dataset/algorithmic/evaluation bias and unintended consequences

KA Core:

10. Formulation of simple machine learning as an optimization problem, such as least squares linear regression or logistic regression
    a. Objective function
    b. Gradient descent
    c. Regularization to avoid overfitting (mathematical formulation)
11. Ensembles of models
    a. Simple weighted majority combination
12. Deep learning
    a. Deep feed-forward networks (intuition only, no mathematics)
    b. Convolutional neural networks (intuition only, no mathematics)
    c. Visualization of learned feature representations from deep nets
    d. Other architectures (generative NN, recurrent NN, transformers, etc.)
13. Performance evaluation
    a. Other metrics for classification (e.g., error, precision, recall)
    b. Performance metrics for regressors
    c. Confusion matrix
    d. Cross-validation
        i. Parameter tuning (grid/random search, via cross-validation)
14. Overview of reinforcement learning methods
15. Two or more applications of machine learning algorithms
    a. E.g., medicine and health, economics, vision, natural language, robotics, game play
16. Ethics for Machine Learning
    a. Continued focus on real data, real scenarios, and case studies (See also: SEP-Context)
    b. Privacy (See also: SEP-Privacy)
    c. Fairness (See also: SEP-Privacy)
    d. Intellectual property
    e. Explainability

Non-core:

17. General statistical-based learning, parameter estimation (maximum likelihood)
18. Supervised learning
    a. Decision trees
    b. Nearest-neighbor classification and regression
    c. Learning simple neural networks / multi-layer perceptrons
    d. Linear regression
    e. Logistic regression
    f. Support vector machines (SVMs) and kernels
    g. Gaussian Processes
19. Overfitting
    a. The curse of dimensionality
    b. Regularization (mathematical computations, L_2 and L_1 regularization)
20. Experimental design

a. Data preparation (e.g., standardization, representation, one-hot encoding)
b. Hypothesis space
c. Biases (e.g., algorithmic, search)
d. Partitioning data: stratification, training set, validation set, test set
e. Parameter tuning (grid/random search, via cross-validation)
f. Performance evaluation
    i. Cross-validation
    ii. Metric: error, precision, recall, confusion matrix
    iii. Receiver operating characteristic (ROC) curve and area under ROC curve

21. Bayesian learning (Cross-Reference AI/Reasoning Under Uncertainty)
    a. Naive Bayes and its relationship to linear models
    b. Bayesian networks
    c. Prior/posterior
    d. Generative models
22. Deep learning
    a. Deep feed-forward networks
    b. Neural tangent kernel and understanding neural network training
    c. Convolutional neural networks
    d. Autoencoders
    e. Recurrent networks
    f. Representations and knowledge transfer
    g. Adversarial training and generative adversarial networks
    h. Attention mechanisms
23. Representations
    a. Manually crafted representations
    b. Basis expansion
    c. Learned representations (e.g., deep neural networks)
24. Unsupervised learning and clustering
    a. K-means
    b. Gaussian mixture models
    c. Expectation maximization (EM)
    d. Self-organizing maps
25. Graph analysis (e.g., PageRank)
26. Semi-supervised learning
27. Graphical models (See also: AI-Probability)
28. Ensembles
    a. Weighted majority
    b. Boosting/bagging
    c. Random forest
    d. Gated ensemble
29. Learning theory
    a. General overview of learning theory / why learning works
    b. VC dimension
    c. Generalization bounds

30. Reinforcement learning
    a. Exploration vs exploitation tradeoff
    b. Markov decision processes
    c. Value and policy iteration
    d. Policy gradient methods
    e. Deep reinforcement learning
    f. Learning from demonstration and inverse RL
31. Explainable / interpretable machine learning
    a. Understanding feature importance (e.g., LIME, Shapley values)
    b. Interpretable models and representations
32. Recommender systems
33. Hardware for machine learning
    a. GPUs / TPUs
34. Application of machine learning algorithms to:
    a. Medicine and health
    b. Economics
    c. Education
    d. Vision
    e. Natural language
    f. Robotics
    g. Game play
    h. Data mining (Cross-reference DM/Data Analytics)
35. Ethics for Machine Learning
    a. Continued focus on real data, real scenarios, and case studies (See also: SEP-Context)
    b. In depth exploration of dataset/algorithmic/evaluation bias, data privacy, and fairness (See also: SEP-Privacy, SEP-Context)
    c. Trust / explainability

Illustrative Learning Outcomes:
1. Describe the differences among the three main styles of learning (supervised, reinforcement, and unsupervised) and determine which is appropriate to a particular problem domain.
2. Differentiate the terms of AI, machine learning, and deep learning.
3. Frame an application as a classification problem, including the available input features and output to be predicted (e.g., identifying alphabetic characters from pixel grid input).
4. Apply two or more simple statistical learning algorithms to a classification task and measure the classifiers' accuracy.
5. Identify overfitting in the context of a problem and learning curves and describe solutions to overfitting.
6. Explain how machine learning works as an optimization/search process.
7. Implement a statistical learning algorithm and the corresponding optimization process to train the classifier and obtain a prediction on new data.
8. Describe the neural network training process and resulting learned representations.

9. Explain proper ML evaluation procedures, including the differences between training and testing performance, and what can go wrong with the evaluation process leading to inaccurate reporting of ML performance.
10. Compare two machine learning algorithms on a dataset, implementing the data preprocessing and evaluation methodology (e.g., metrics and handling of train/test splits) from scratch.
11. Visualize the training progress of a neural network through learning curves in a well-established toolkit (e.g., TensorBoard) and visualize the learned features of the network.
12. Compare and contrast several learning techniques (e.g., decision trees, logistic regression, naive Bayes, neural networks, and belief networks), providing examples of when each strategy is superior.
13. Evaluate the performance of a simple learning system on a real-world dataset.
14. Characterize the state of the art in learning theory, including its achievements and shortcomings.
15. Explain the problem of overfitting, along with techniques for detecting and managing the problem.
16. Explain the triple tradeoff among the size of a hypothesis space, the size of the training set, and performance accuracy.
17. Given a real-world application of machine learning, describe ethical issues regarding the choices of data, preprocessing steps, algorithm selection, and visualization/presentation of results.

AI-SEP: Applications and Societal Impact

Note: There is substantial benefit to studying applications and ethics/fairness/trust/explainability in a curriculum alongside the methods and theory that they apply to, rather than covering ethics in a separate, dedicated class session. Whenever possible, study of these topics should be integrated alongside other modules, such as exploring how decision trees could be applied to a specific problem in environmental sustainability such as land use allocation, then assessing the social/environmental/ethical implications of doing so.

CS Core:

1. At least one application of AI to a specific problem and field, such as medicine, health, sustainability, social media, economics, education, robotics, etc. (choose at least one for the CS Core).
    a. Formulating and evaluating a specific application as an AI problem
        i. How to deal with underspecified or ill-posed problems
    b. Data availability/scarcity and cleanliness
        i. Basic data cleaning and preprocessing
        ii. Data set bias
    c. Algorithmic bias
    d. Evaluation bias
    e. Assessment of societal implications of the application
2. Deployed deep generative models
    a. High-level overview of deep image generative models (e.g., as of 2023, DALL-E, Midjourney, Stable Diffusion, etc.), their uses, and their shortcomings/pitfalls.
    b. High-level overview of large language models (e.g., as of 2023, ChatGPT, Bard, etc.), their uses, and their shortcomings/pitfalls.
3. Overview of societal impact of AI

    a. Ethics (See also: SEP-Context)
    b. Fairness (See also: SEP-Privacy, SEP-DEIA)
    c. Trust/explainability (See also: SEP-Context)
    d. Privacy and usage of training data (See also: SEP-Privacy)
    e. Human autonomy and oversight/regulations/legal requirements (See also: SEP-Context)
    f. Sustainability (See also: SEP-Sustainability)

KA Core:

4. One or more additional applications of AI to a broad set of problems and diverse fields, such as medicine, health, sustainability, social media, economics, education, robotics, etc. (choose a different area from that chosen for the CS Core).
    a. Formulating and evaluating a specific application as an AI problem
        i. How to deal with underspecified or ill-posed problems
    b. Data availability/scarcity and cleanliness
        i. Basic data cleaning and preprocessing
        ii. Data set bias
    c. Algorithmic bias
    d. Evaluation bias
    e. Assessment of societal implications of the application
5. Additional depth on deployed deep generative models
    a. Introduction to how deep image generative models work, (e.g., as of 2023, DALL-E, Midjourney, Stable Diffusion) including discussion of attention
    b. Introduction to how large language models work, (e.g., as of 2023, ChatGPT, Bard) including discussion of attention
    c. Idea of foundational models, how to use them, and the benefits/issues with training them from big data
6. Analysis and discussion of the societal impact of AI
    a. Ethics (See also: SEP-Context)
    b. Fairness (See also: SEP-Privacy, SEP-DEIA)
    c. Trust/explainability (See also: SEP-Context)
    d. Privacy and usage of training data (See also: SEP-Privacy)
    e. Human autonomy and oversight/regulations/legal requirements (See also: SEP-Context)
    f. Sustainability (See also: SEP-Sustainability)

Illustrative Learning Outcomes:

1. Given a real-world application domain and problem, formulate an AI solution to it, identifying proper data/input, preprocessing, representations, AI techniques, and evaluation metrics/methodology.
2. Analyze the societal impact of one or more specific real-world AI applications, identifying issues regarding ethics, fairness, bias, trust, and explainability.
3. Describe some of the failure modes of current deep generative models for language or images, and how this could affect their use in an application.

AI-LRR: Logical Representation and Reasoning

Non-core:

1. Review of propositional and predicate logic (See also: MSF-Discrete)
2. Resolution and theorem proving (propositional logic only)
    a. Forward chaining, backward chaining
3. Knowledge representation issues
    a. Description logics
    b. Ontology engineering
4. Semantic web
5. Non-monotonic reasoning (e.g., non-classical logics, default reasoning)
6. Argumentation
7. Reasoning about action and change (e.g., situation and event calculus)
8. Temporal and spatial reasoning
9. Logic programming
    a. Prolog, Answer Set Programming
10. Rule-based Expert Systems
11. Semantic networks
12. Model-based and Case-based reasoning

Illustrative Learning Outcomes:
1. Translate a natural language (e.g., English) sentence into a predicate logic statement.
2. Convert a logic statement into clausal form.
3. Apply resolution to a set of logic statements to answer a query.
4. Compare and contrast the most common models used for structured knowledge representation, highlighting their strengths and weaknesses.
5. Identify the components of non-monotonic reasoning and its usefulness as a representational mechanism for belief systems.
6. Compare and contrast the basic techniques for representing uncertainty.
7. Compare and contrast the basic techniques for qualitative representation.
8. Apply situation and event calculus to problems of action and change.
9. Explain the distinction between temporal and spatial reasoning, and how they interrelate.
10. Explain the difference between rule-based, case-based, and model-based reasoning techniques.
11. Define the concept of a planning system and how it differs from classical search techniques.
12. Describe the differences between planning as search, operator-based planning, and propositional planning, providing examples of domains where each is most applicable.
13. Explain the distinction between monotonic and non-monotonic inference.

AI-Probability: Probabilistic Representation and Reasoning

Non-core:
1. Conditional Independence review
2. Knowledge representations
    a. Bayesian Networks
        i. Exact inference and its complexity
        ii. Markov blankets and d-separation
        iii. Randomized sampling (Monte Carlo) methods (e.g., Gibbs sampling)
    b. Markov Networks

    c. Relational probability models
    d. Hidden Markov Models
3. Decision Theory
    a. Preferences and utility functions
    b. Maximizing expected utility
    c. Game theory

Illustrative Learning Outcomes:

1. Compute the probability of a hypothesis given the evidence in a Bayesian network.
2. Explain how conditional independence assertions allow for greater efficiency of probabilistic systems.
3. Identify examples of knowledge representations for reasoning under uncertainty.
4. State the complexity of exact inference. Identify methods for approximate inference.
5. Design and implement at least one knowledge representation for reasoning under uncertainty.
6. Describe the complexities of temporal probabilistic reasoning.
7. Design and implement an HMM as one example of a temporal probabilistic system.
8. Describe the relationship between preferences and utility functions.
9. Explain how utility functions and probabilistic reasoning can be combined to make rational decisions.

AI-Planning: Planning

Non-core:

1. Review of propositional and first-order logic
2. Planning operators and state representations
3. Total order planning
4. Partial-order planning
5. Plan graphs and GraphPlan
6. Hierarchical planning
7. Planning languages and representations
    a. PDDL
8. Multi-agent planning
9. MDP-based planning
10. Interconnecting planning, execution, and dynamic replanning
    a. Conditional planning
    b. Continuous planning
    c. Probabilistic planning

Illustrative Learning Outcomes:

1. Construct the state representation, goal, and operators for a given planning problem.
2. Encode a planning problem in PDDL and use a planner to solve it.
3. Given a set of operators, initial state, and goal state, draw the partial-order planning graph and include ordering constraints to resolve all conflicts.
4. Construct the complete planning graph for GraphPlan to solve a given problem.

AI-Agents: Agents and Cognitive Systems

Non-core:

1. Agent architectures (e.g., reactive, layered, cognitive)
2. Agent theory (including mathematical formalisms)
3. Rationality, Game Theory
   a. Decision-theoretic agents
   b. Markov decision processes (MDP)
   c. Bandit algorithms
4. Software agents, personal assistants, and information access
   a. Collaborative agents
   b. Information-gathering agents
   c. Believable agents (synthetic characters, modeling emotions in agents)
5. Learning agents
6. Cognitive systems
   a. Cognitive architectures (e.g., ACT-R, SOAR, ICARUS, FORR)
   b. Capabilities (e.g., perception, decision making, prediction, knowledge maintenance)
   c. Knowledge representation, organization, utilization, acquisition, and refinement
   d. Applications and evaluation of cognitive systems
7. Multi-agent systems
   a. Collaborating agents
   b. Agent teams
   c. Competitive agents (e.g., auctions, voting)
   d. Swarm systems and biologically inspired models
   e. Multi-agent learning
8. Human-agent interaction (See also: HCI-User, HCI-Accessibility)
   a. Communication methodologies (verbal and non-verbal)
   b. Practical issues
   c. Applications
      i. Trading agents, supply chain management
      ii. Ethical issues of AI interactions with humans
      iii. Regulation and legal requirements of AI systems for interacting with humans

Illustrative Learning Outcomes:

1. Characterize and contrast the standard agent architectures.
2. Describe the applications of agent theory to domains such as software agents, personal assistants, and believable agents, and discuss associated ethical implications.
3. Describe the primary paradigms used by learning agents.
4. Demonstrate using appropriate examples how multi-agent systems support agent interaction.
5. Construct an intelligent agent using a well-established cognitive architecture (ACT-R, SOAR) for solving a specific problem.

AI-NLP: Natural Language Processing

Non-core:

1. Deterministic and stochastic grammars
2. Parsing algorithms
    a. CFGs and chart parsers (e.g., CYK)
    b. Probabilistic CFGs and weighted CYK
3. Representing meaning/Semantics
    a. Logic-based knowledge representations
    b. Semantic roles
    c. Temporal representations
    d. Beliefs, desires, and intentions
4. Corpus-based methods
5. N-grams and HMMs
6. Smoothing and backoff
7. Examples of use: POS tagging and morphology
8. Information retrieval (See also: DM-Unstructured)
    a. Vector space model
        i. TF & IDF
    b. Precision and recall
9. Information extraction
10. Language translation
11. Text classification, categorization
    a. Bag of words model
12. Deep learning for NLP (See also: AI-ML)
    a. RNNs
    b. Transformers
    c. Multi-modal embeddings (e.g., images + text)
    d. Generative language models

Illustrative Learning Outcomes:
1. Define and contrast deterministic and stochastic grammars, providing examples to show the adequacy of each.
2. Simulate, apply, or implement classic and stochastic algorithms for parsing natural language.
3. Identify the challenges of representing meaning.
4. List the advantages of using standard corpora. Identify examples of current corpora for a variety of NLP tasks.
5. Identify techniques for information retrieval, language translation, and text classification.
6. Implement a TF/IDF transform, use it to extract features from a corpus, and train an off-the-shelf machine learning algorithm using those features to do text classification.

AI-Robotics: Robotics

(See also: SPD-Robot)
Non-core:
1. Overview: problems and progress
    a. State-of-the-art robot systems, including their sensors and an overview of their sensor processing

   b. Robot control architectures, e.g., deliberative vs reactive control and Braitenberg vehicles
   c. World modeling and world models
   d. Inherent uncertainty in sensing and in control
2. Sensors and effectors
   a. Sensors: e.g., LIDAR, sonar, vision, depth, stereoscopic, event cameras, microphones, haptics,
   b. Effectors: e.g., wheels, arms, grippers
3. Coordinate frames, translation, and rotation (2D and 3D)
4. Configuration space and environmental maps
5. Interpreting uncertain sensor data
6. Localization and mapping
7. Navigation and control
8. Forward and inverse kinematics
9. Motion path planning and trajectory optimization
10. Manipulation and grasping
11. Joint control and dynamics
12. Vision-based control
13. Multiple-robot coordination and collaboration
14. Human-robot interaction (See also: HCI-User, HCI-Accessibility)
    a. Shared workspaces
    b. Human-robot teaming and physical HRI
    c. Social assistive robots
    d. Motion/task/goal prediction
    e. Collaboration and communication (explicit vs implicit, verbal or symbolic vs non-verbal or visual)
    f. Trust
15. Applications and Societal, Economic, and Ethical Issues
    a. Societal, economic, right-to-work implications
    b. Ethical and privacy implications of robotic applications
    c. Liability in autonomous robotics
    d. Autonomous weapons and ethics
    e. Human oversight and control

Illustrative Learning Outcomes:

(Note: Due to the expense of robot hardware, all of these could be done in simulation or with low-cost educational robotic platforms.)

1. List capabilities and limitations of today's state-of-the-art robot systems, including their sensors and the crucial sensor processing that informs those systems.
2. Integrate sensors, actuators, and software into a robot designed to undertake a specific task.
3. Program a robot to accomplish simple tasks using deliberative, reactive, and/or hybrid control architectures.
4. Implement fundamental motion planning algorithms within a robot configuration space.
5. Characterize the uncertainties associated with common robot sensors and actuators; articulate strategies for mitigating these uncertainties.
6. List the differences among robots' representations of their external environment, including their strengths and shortcomings.

7. Compare and contrast at least three strategies for robot navigation within known and/or unknown environments, including their strengths and shortcomings.
8. Describe at least one approach for coordinating the actions and sensing of several robots to accomplish a single task.
9. Compare and contrast a multi-robot coordination and a human-robot collaboration approach and attribute their differences to differences between the problem settings.
10. Analyze the societal, economic, and ethical issues of a real-world robotics application.

AI-Vision: Perception and Computer Vision

Non-core:

1. Computer vision
   a. Image acquisition, representation, processing, and properties
   b. Shape representation, object recognition, and segmentation
   c. Motion analysis
   d. Generative models
2. Audio and speech recognition
3. Touch and proprioception
4. Other modalities (e.g., olfaction)
5. Modularity in recognition
6. Approaches to pattern recognition (See also: AI-ML)
   a. Classification algorithms and measures of classification quality
   b. Statistical techniques
   c. Deep learning techniques

Illustrative Learning Outcomes:

1. Summarize the importance of image and object recognition in AI and indicate several significant applications of this technology.
2. List at least three image-segmentation approaches, such as thresholding, edge-based and region-based algorithms, along with their defining characteristics, strengths, and weaknesses.
3. Implement 2d object recognition based on contour-based and/or region-based shape representations.
4. Distinguish the goals of sound-recognition, speech-recognition, and speaker-recognition and identify how the raw audio signal will be handled differently in each of these cases.
5. Provide at least two examples of a transformation of a data source from one sensory domain to another, e.g., tactile data interpreted as single-band 2d images.
6. Implement a feature-extraction algorithm on real data, e.g., an edge or corner detector for images or vectors of Fourier coefficients describing a short slice of audio signal.
7. Implement an algorithm combining features into higher-level percepts, e.g., a contour or polygon from visual primitives or phoneme hypotheses from an audio signal.
8. Implement a classification algorithm that segments input percepts into output categories and quantitatively evaluates the resulting classification.
9. Evaluate the performance of the underlying feature-extraction, relative to at least one alternative possible approach (whether implemented or not) in its contribution to the classification task (8), above.

10. Describe at least three classification approaches, their prerequisites for applicability, their strengths, and their shortcomings.
11. Implement and evaluate a deep learning solution to problems in computer vision, such as object or scene recognition.

Professional Dispositions

- **Meticulousness:** Since attention must be paid to details when implementing AI and machine learning algorithms, students must be meticulous about detail.
- **Persistence:** AI techniques often operate in partially observable environments and optimization processes may have cascading errors from multiple iterations. Getting AI techniques to work predictably takes trial and error, and repeated effort. These call for persistence on the part of the student.
- **Inventive:** Applications of AI involve creative problem formulation and application of AI techniques, while balancing application requirements and societal and ethical issues.
- **Responsible:** Applications of AI can have significant impacts on society, affecting both individuals and large populations. This calls for students to understand the implications of work in AI to society, and to make responsible choices for when and how to apply AI techniques.

Mathematics Requirements

Required:

- Algebra
- Precalculus
- Discrete Math (See also: MSF-Discrete)
  - sets, relations, functions, graphs
  - predicate and first-order logic, logic-based proofs
- Linear Algebra (See also: MSF-Linear)
  - Matrix operations, matrix algebra
  - Basis sets
- Probability and Statistics (See also: MSF-Statistics)
  - Basic probability theory, conditional probability, independence
  - Bayes theorem and applications of Bayes theorem
  - Expected value, basic descriptive statistics, distributions
  - Basic summary statistics and significance testing
  - All should be applied to real decision-making examples with real data, not “textbook” examples.

Desirable:

- Calculus-based probability and statistics
- Calculus: single-variable and partial derivatives
- Other topics in probability and statistics
  - Hypothesis testing, data resampling, experimental design techniques
- Optimization
- Linear algebra (all other topics)

Course Packaging Suggestions

Artificial Intelligence to include the following:
- AI-Introduction (4 hours)
- AI-Search (9 hours)
- AI-KRR (4 hours)
- AI-ML (12 hours)
- AI-Probability (5 hours)
- AI-SEP (4 hours –integrated throughout the course)

Prerequisites:
- SDF-Fundamentals
- SDF-Data-Structures
- SDF-Algorithms
- MSF-Discrete
- MSF-Probability

Course objective: A student who completes this course should understand the basic areas of AI and be able to understand, develop, and apply techniques in each. They should be able to solve problems using search techniques, basic Bayesian reasoning, and simple machine learning methods. They should understand the various applications of AI and associated ethical and societal implications.

Machine Learning to include the following:
- AI-ML (32 hours)
- AI-KRR (4 hours)
- AI-NLP (4 hours – selected topics, e.g., TF-IDF, bag of words, and text classification)
- AI-SEP (4 hours – should be integrated throughout the course)

Prerequisites:
- SDF-Fundamentals
- SDF-Data-Structures
- SDF-Algorithms
- MSF-Discrete
- MSF-Probability
- MSF-Statistics
- MSF-Linear (optional)

Course objective: A student who completes this course should be able to understand, develop, and apply mechanisms for supervised, unsupervised, and reinforcement learning. They should be able to select the proper machine learning algorithm for a problem, preprocess the data appropriately, apply proper evaluation techniques, and explain how to interpret the resulting models, including the model's shortcomings. They should be able to identify and compensate for biased data sets and other sources of error and be able to explain ethical and societal implications of their application of machine learning to practical problems.

Robotics to include the following:
- AI-Robotics (25 hours)

- SPD-Robot (4 hours – focusing on hardware, constraints/considerations, and software architectures; other topics in SPD/Robot Platforms that overlap with AI/Robotics)
- AI-Search (4 hours – selected topics well-integrated with robotics, e.g., A* and path search)
- AI-ML (6 hours – selected topics well-integrated with robotics, e.g., neural networks for object recognition)
- AI-SEP (3 hours – integrated throughout the course; robotics is already a huge application, so this really should focus on societal impact and specific robotic applications).

Prerequisites:
- SDF-Fundamentals
- SDF-Data-Structures
- SDF-Algorithms
- MSF-Linear

Course objective: A student who completes this course should be able to understand and use robotic techniques to perceive the world using sensors, localize the robot based on features and a map, and plan paths and navigate in the world in simple robot applications. They should understand and be able to apply simple computer vision, motion planning, and forward and inverse kinematics techniques.

Introduction to Data Science to include the following:
- GIT-Visualization (6 hours) – types of visualization, libraries, foundations
- GIT-SEP (2 hours) – ethically responsible visualization
- DM-Core (2 hours) – Parallel and distributed processing (MapReduce, cloud frameworks, etc.)–
- DM-Modeling (2 hours) – Graph representations, entity resolution
- DM-Querying (4 hours) – SQL, query formation
- DM-NoSQL (2 hours) – Graph DBs, data lakes, data consistency
- DM-Security (2 hours) – privacy, personally identifying information and its protection
- DM-Analytics (1 hour) – exploratory data techniques, data science lifecycle
- DM-SEP (2 hours) – Data provenance
- AI-ML (15 hours) – Data preprocessing, missing data imputation, supervised/semi-supervised/unsupervised learning, text analysis, graph analysis and PageRank, experimental methodology, evaluation, and ethics
- AI-SEP (3 hours) – Applications specific to data science, interspersed throughout the course
- MSF-Statistics (3 hours) – Statistical analysis, hypothesis testing, experimental design

Prerequisites:
- SDF-Fundamentals

Course objective: A student who completes this course should be able to formulate questions as data analysis problems, understand and use statistical techniques to achieve that analysis from real data, apply visualization techniques to convey the results, and analyze the ethical and societal implications of data science applications. Students should also be able to understand and effectively use data management techniques for preprocessing, storage, security, and retrieval of data in current systems.

Committee

Chair: Eric Eaton, University of Pennsylvania, Philadelphia, PA, USA

Members:

- Zachary Dodds, Harvey Mudd College, Claremont, CA, USA
- Susan L. Epstein, Hunter College and The Graduate Center of The City University of New York, New York, NY, USA
- Laura Hiatt, US Naval Research Laboratory, Washington, DC, USA
- Amruth N. Kumar, Ramapo College of New Jersey, Mahwah, NJ, USA
- Peter Norvig, Google, Mountain View, CA, USA
- Meinolf Sellmann, GE Research, Niskayuna, NY, USA
- Reid Simmons, Carnegie Mellon University, Pittsburgh, PA, USA

Contributors:

- Nate Derbinsky, Northeastern University, Boston, MA, USA
- Eugene Freuder, Insight Centre for Data Analytics, University College Cork, Cork, Ireland
- Ashok Goel, Georgia Institute of Technology, Atlanta, GA, USA
- Claudia Schulz, Thomson Reuters, Zurich, Switzerland