

Intelligent Systems (IS)

Artificial intelligence (AI) is the study of solutions for problems that are difficult or impractical to solve with traditional methods. It is used pervasively in support of everyday applications such as email, word-processing and search, as well as in the design and analysis of autonomous agents that perceive their environment and interact rationally with the environment.

The solutions rely on a broad set of general and specialized knowledge representation schemes, problem solving mechanisms and learning techniques. They deal with sensing (e.g., speech recognition, natural language understanding, computer vision), problem-solving (e.g., search, planning), and acting (e.g., robotics) and the architectures needed to support them (e.g., agents, multi-agents). The study of Artificial Intelligence prepares the student to determine when an AI approach is appropriate for a given problem, identify the appropriate representation and reasoning mechanism, and implement and evaluate it.

IS. Intelligent Systems (10 Core-Tier2 hours)

	Core-Tier1 hours	Core-Tier2 hours	Includes Electives
IS/Fundamental Issues		1	Y
IS/Basic Search Strategies		4	N
IS/Basic Knowledge Representation and Reasoning		3	N
IS/Basic Machine Learning		2	N
IS/Advanced Search			Y
IS/Advanced Representation and Reasoning			Y
IS/Reasoning Under Uncertainty			Y
IS/Agents			Y
IS/Natural Language Processing			Y
IS/Advanced Machine Learning			Y
IS/Robotics			Y
IS/Perception and Computer Vision			Y

IS/Fundamental Issues

[1 Core-Tier2 hours]

Topics:

- Overview of AI problems, examples of successful recent AI applications
- What is intelligent behavior?
 - The Turing test
 - Rational versus non-rational reasoning
- Problem characteristics
 - Fully versus partially observable
 - Single versus multi-agent
 - Deterministic versus stochastic
 - Static versus dynamic
 - Discrete versus continuous
- Nature of agents
 - Autonomous versus semi-autonomous
 - Reflexive, goal-based, and utility-based
 - The importance of perception and environmental interactions
- Philosophical and ethical issues. [elective]

Learning Outcomes:

1. Describe Turing test and the “Chinese Room” thought experiment. [Familiarity]
2. Differentiate between the concepts of optimal reasoning/behavior and human-like reasoning/behavior. [Familiarity]
3. Determine the characteristics of a given problem that an intelligent system must solve. [Assessment]

IS/Basic Search Strategies

[4 Core-Tier2 hours]

Cross-reference AL/Basic Analysis, AL/Algorithmic Strategies, AL/Fundamental Data Structures and Algorithms

Topics:

- Problem spaces (states, goals and operators), problem solving by search
- Factored representation (factoring state into variables)
- Uninformed search (breadth-first, depth-first, depth-first with iterative deepening)
- Heuristics and informed search (hill-climbing, generic best-first, A*)
- Space and time efficiency of search
- Two-player games (introduction to minimax search)
- Constraint satisfaction (backtracking and local search methods)

Learning Outcomes:

1. Formulate an efficient problem space for a problem expressed in natural language (e.g., English) in terms of initial and goal states, and operators. [Usage]
2. Describe the role of heuristics and describe the trade-offs among completeness, optimality, time complexity, and space complexity. [Familiarity]
3. Describe the problem of combinatorial explosion of search space and its consequences. [Familiarity]
4. Select and implement an appropriate uninformed search algorithm for a problem, and characterize its time and space complexities. [Usage]
5. Select and implement an appropriate informed search algorithm for a problem by designing the necessary heuristic evaluation function. [Usage]
6. Evaluate whether a heuristic for a given problem is admissible/can guarantee optimal solution. [Assessment]
7. Formulate a problem specified in natural language (e.g., English) as a constraint satisfaction problem and implement it using a chronological backtracking algorithm or stochastic local search. [Usage]
8. Compare and contrast basic search issues with game playing issues. [Familiarity]

IS/Basic Knowledge Representation and Reasoning

[3 Core-Tier2 hours]

Topics:

- Review of propositional and predicate logic (cross-reference DS/Basic Logic)
- Resolution and theorem proving (propositional logic only)
- Forward chaining, backward chaining
- Review of probabilistic reasoning, Bayes theorem (cross-reference with DS/Discrete Probability)

Learning Outcomes:

1. Translate a natural language (e.g., English) sentence into predicate logic statement. [Usage]
2. Convert a logic statement into clause form. [Usage]
3. Apply resolution to a set of logic statements to answer a query. [Usage]
4. Make a probabilistic inference in a real-world problem using Bayes' theorem to determine the probability of a hypothesis given evidence. [Usage]

IS/Basic Machine Learning

[2 Core-Tier2 hours]

Topics:

- Definition and examples of broad variety of machine learning tasks, including classification
- Inductive learning
- Simple statistical-based learning, such as Naive Bayesian Classifier, decision trees
- The over-fitting problem
- Measuring classifier accuracy

Learning Outcomes:

1. List the differences among the three main styles of learning: supervised, reinforcement, and unsupervised. [Familiarity]
2. Identify examples of classification tasks, including the available input features and output to be predicted. [Familiarity]
3. Explain the difference between inductive and deductive learning. [Familiarity]

4. Describe over-fitting in the context of a problem. [Familiarity]
5. Apply the simple statistical learning algorithm such as Naive Bayesian Classifier to a classification task and measure the classifier's accuracy. [Usage]

IS/Advanced Search

[Elective]

Note that the general topics of Branch-and-bound and Dynamic Programming are listed in AL/Algorithmic Strategies.

Topics:

- Constructing search trees, dynamic search space, combinatorial explosion of search space
- Stochastic search
 - Simulated annealing
 - Genetic algorithms
 - Monte-Carlo tree search
- Implementation of A* search, beam search
- Minimax search, alpha-beta pruning
- Expectimax search (MDP-solving) and chance nodes

Learning Outcomes:

1. Design and implement a genetic algorithm solution to a problem. [Usage]
2. Design and implement a simulated annealing schedule to avoid local minima in a problem. [Usage]
3. Design and implement A*/beam search to solve a problem. [Usage]
4. Apply minimax search with alpha-beta pruning to prune search space in a two-player game. [Usage]
5. Compare and contrast genetic algorithms with classic search techniques. [Assessment]
6. Compare and contrast various heuristic searches vis-a-vis applicability to a given problem. [Assessment]

IS/Advanced Representation and Reasoning

[Elective]

Topics:

- Knowledge representation issues
 - Description logics
 - Ontology engineering
- Non-monotonic reasoning (e.g., non-classical logics, default reasoning)
- Argumentation
- Reasoning about action and change (e.g., situation and event calculus)
- Temporal and spatial reasoning
- Rule-based Expert Systems
- Semantic networks
- Model-based and Case-based reasoning
- Planning:
 - Partial and totally ordered planning
 - Plan graphs
 - Hierarchical planning
 - Planning and execution including conditional planning and continuous planning
 - Mobile agent/Multi-agent planning

Learning Outcomes:

1. Compare and contrast the most common models used for structured knowledge representation, highlighting their strengths and weaknesses. [Assessment]
2. Identify the components of non-monotonic reasoning and its usefulness as a representational mechanism for belief systems. [Familiarity]
3. Compare and contrast the basic techniques for representing uncertainty. [Assessment]
4. Compare and contrast the basic techniques for qualitative representation. [Assessment]
5. Apply situation and event calculus to problems of action and change. [Usage]
6. Explain the distinction between temporal and spatial reasoning, and how they interrelate. [Familiarity]
7. Explain the difference between rule-based, case-based and model-based reasoning techniques. [Familiarity]
8. Define the concept of a planning system and how it differs from classical search techniques. [Familiarity]
9. Describe the differences between planning as search, operator-based planning, and propositional planning, providing examples of domains where each is most applicable. [Familiarity]
10. Explain the distinction between monotonic and non-monotonic inference. [Familiarity]

IS/Reasoning Under Uncertainty

[Elective]

Topics:

- Review of basic probability (cross-reference DS/Discrete Probability)
- Random variables and probability distributions
 - Axioms of probability
 - Probabilistic inference
 - Bayes' Rule
- Conditional Independence
- Knowledge representations
 - Bayesian Networks
 - Exact inference and its complexity
 - Randomized sampling (Monte Carlo) methods (e.g. Gibbs sampling)
 - Markov Networks
 - Relational probability models
 - Hidden Markov Models
- Decision Theory
 - Preferences and utility functions
 - Maximizing expected utility

Learning Outcomes:

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

IS/Agents

[Elective]

Cross-reference HCI/Collaboration and Communication

Topics:

- Definitions of agents
- Agent architectures (e.g., reactive, layered, cognitive)
- Agent theory
- Rationality, game theory
 - Decision-theoretic agents
 - Markov decision processes (MDP)
- Software agents, personal assistants, and information access
 - Collaborative agents
 - Information-gathering agents
 - Believable agents (synthetic characters, modeling emotions in agents)
- Learning agents
- Multi-agent systems
 - Collaborating agents
 - Agent teams
 - Competitive agents (e.g., auctions, voting)
 - Swarm systems and biologically inspired models

Learning Outcomes:

1. List the defining characteristics of an intelligent agent. [Familiarity]
2. Characterize and contrast the standard agent architectures. [Assessment]
3. Describe the applications of agent theory to domains such as software agents, personal assistants, and believable agents. [Familiarity]
4. Describe the primary paradigms used by learning agents. [Familiarity]
5. Demonstrate using appropriate examples how multi-agent systems support agent interaction. [Usage]

IS/Natural Language Processing

[Elective]

Cross-reference HCI/New Interactive Technologies

Topics:

- Deterministic and stochastic grammars
- Parsing algorithms
 - CFGs and chart parsers (e.g. CYK)
 - Probabilistic CFGs and weighted CYK
- Representing meaning / Semantics
 - Logic-based knowledge representations
 - Semantic roles
 - Temporal representations
 - Beliefs, desires, and intentions
- Corpus-based methods
- N-grams and HMMs
- Smoothing and backoff

- Examples of use: POS tagging and morphology
- Information retrieval (Cross-reference IM/Information Storage and Retrieval)
 - Vector space model
 - TF & IDF
 - Precision and recall
- Information extraction
- Language translation
- Text classification, categorization
 - Bag of words model

Learning Outcomes:

1. Define and contrast deterministic and stochastic grammars, providing examples to show the adequacy of each. [Assessment]
2. Simulate, apply, or implement classic and stochastic algorithms for parsing natural language. [Usage]
3. Identify the challenges of representing meaning. [Familiarity]
4. List the advantages of using standard corpora. Identify examples of current corpora for a variety of NLP tasks. [Familiarity]
5. Identify techniques for information retrieval, language translation, and text classification. [Familiarity]

IS/Advanced Machine Learning

[Elective]

Topics:

- Definition and examples of broad variety of machine learning tasks
- General statistical-based learning, parameter estimation (maximum likelihood)
- Inductive logic programming (ILP)
- Supervised learning
 - Learning decision trees
 - Learning neural networks
 - Support vector machines (SVMs)
- Ensembles
- Nearest-neighbor algorithms
- Unsupervised Learning and clustering
 - EM
 - K-means
 - Self-organizing maps
- Semi-supervised learning
- Learning graphical models (Cross-reference IS/Reasoning under Uncertainty)
- Performance evaluation (such as cross-validation, area under ROC curve)
- Learning theory
- The problem of overfitting, the curse of dimensionality
- Reinforcement learning
 - Exploration vs. exploitation trade-off
 - Markov decision processes
 - Value and policy iteration
- Application of Machine Learning algorithms to Data Mining (cross-reference IM/Data Mining)

Learning Outcomes:

1. Explain the differences among the three main styles of learning: supervised, reinforcement, and unsupervised. [Familiarity]
2. Implement simple algorithms for supervised learning, reinforcement learning, and unsupervised learning. [Usage]
3. Determine which of the three learning styles is appropriate to a particular problem domain. [Usage]
4. Compare and contrast each of the following techniques, providing examples of when each strategy is superior: decision trees, neural networks, and belief networks. [Assessment]
5. Evaluate the performance of a simple learning system on a real-world dataset. [Assessment]
6. Characterize the state of the art in learning theory, including its achievements and its shortcomings. [Familiarity]
7. Explain the problem of overfitting, along with techniques for detecting and managing the problem. [Usage]

IS/Robotics

[Elective]

Topics:

- Overview: problems and progress
 - State-of-the-art robot systems, including their sensors and an overview of their sensor processing
 - Robot control architectures, e.g., deliberative vs. reactive control and Braitenberg vehicles
 - World modeling and world models
 - Inherent uncertainty in sensing and in control
- Configuration space and environmental maps
- Interpreting uncertain sensor data
- Localizing and mapping
- Navigation and control
- Motion planning
- Multiple-robot coordination

Learning Outcomes:

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. [Familiarity]
2. Integrate sensors, actuators, and software into a robot designed to undertake some task. [Usage]
3. Program a robot to accomplish simple tasks using deliberative, reactive, and/or hybrid control architectures. [Usage]
4. Implement fundamental motion planning algorithms within a robot configuration space. [Usage]
5. Characterize the uncertainties associated with common robot sensors and actuators; articulate strategies for mitigating these uncertainties. [Familiarity]
6. List the differences among robots' representations of their external environment, including their strengths and shortcomings. [Familiarity]
7. Compare and contrast at least three strategies for robot navigation within known and/or unknown environments, including their strengths and shortcomings. [Assessment]
8. Describe at least one approach for coordinating the actions and sensing of several robots to accomplish a single task. [Familiarity]

IS/Perception and Computer Vision

[Elective]

Topics:

- Computer vision
 - Image acquisition, representation, processing and properties
 - Shape representation, object recognition and segmentation
 - Motion analysis
- Audio and speech recognition
- Modularity in recognition
- Approaches to pattern recognition (cross-reference IS/Advanced Machine Learning)
 - Classification algorithms and measures of classification quality
 - Statistical techniques

Learning Outcomes:

1. Summarize the importance of image and object recognition in AI and indicate several significant applications of this technology. [Familiarity]
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. [Familiarity]
3. Implement 2d object recognition based on contour- and/or region-based shape representations. [Usage]
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. [Familiarity]
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. [Familiarity]
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. [Usage]
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. [Usage]
8. Implement a classification algorithm that segments input percepts into output categories and quantitatively evaluates the resulting classification. [Usage]
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. [Assessment]
10. Describe at least three classification approaches, their prerequisites for applicability, their strengths, and their shortcomings. [Familiarity]

Computer Science Curricula 2013

Curriculum Guidelines for
Undergraduate Degree Programs
in Computer Science

December 20, 2013

The Joint Task Force on Computing Curricula
Association for Computing Machinery (ACM)
IEEE Computer Society

A Cooperative Project of



Association for
Computing Machinery

Advancing Computing as a Science & Profession



IEEE

IEEE
computer
society

Computer Science Curricula 2013

Curriculum Guidelines for Undergraduate Degree Programs in Computer Science

December 20, 2013

The Joint Task Force on Computing Curricula
Association for Computing Machinery (ACM)
IEEE Computer Society



Association for
Computing Machinery

Advancing Computing as a Science & Profession



IEEE

IEEE
computer
society

Computer Science Curricula 2013

**Copyright © 2013 by ACM and IEEE.
All rights reserved.**

Copyright and Reprint Permissions: Permission is granted to use these curriculum guidelines for the development of educational materials and programs. Other use requires specific permission. Permission requests should be addressed to: ACM Permissions Dept. at permissions@acm.org or to the IEEE Copyrights Manager at copyrights@ieee.org.

ISBN: 978-1-4503-2309-3

DOI: 10.1145/2534860

Web link: <http://dx.doi.org/10.1145/2534860>

ACM Order Number: 999133

Additional copies may be ordered from:

IEEE Computer Society
Customer Service Center
10662 Los Vaqueros
P.O. Box 3014
Los Alamitos, CA 90720-1314

Tel: + 1 800 272 6657
Fax: + 1 714 821 4641
<http://computer.org/cspress>
csbook@computer.org

IEEE Service Center IEEE
445 Hoes Lane
P.O. Box 1331
Piscataway, NJ 08855-1331

Tel: + 1 732 981 0060
Fax: + 1 732 981 9667
<http://shop.ieee.org/store/>
customerservice@ieee.org

IEEE Computer Society
Asia/Pacific Office
Watanabe Bldg., 1-4-2
Minami-Aoyama
Minato-ku, Tokyo 107-0062
JAPAN

Tel: + 81 3 3408 3118
Fax: + 81 3 3408 3553
tokyo.ofc@computer.org

ACM Order Department
P.O. Box 30777
New York, NY 10087-0777

1-800-342-6626
1-212-626-0500 (outside U.S.)
orders@acm.org

Cover art by Robert Vizzini

Printed in the United States of America

Sponsoring Societies

This report was made possible by
financial support from the following societies:

ACM
IEEE Computer Society

The CS2013 Final Report has been endorsed by ACM and the IEEE Computer Society.



Association for
Computing Machinery

Advancing Computing as a Science & Profession



IEEE

IEEE
**computer
society**

Computer Science Curricula 2013

Final Report

December 2013

The Joint Task Force on Computing Curricula
Association for Computing Machinery
IEEE-Computer Society