Engineering and Applied Science Programs for Professionals
Whiting School of Engineering
Johns Hopkins University
685.621 Algorithms for Data Science
Introduction

This document provides a rollup of the course introduction, the Data Science curriculum and how this course ties in with data science. Additionally, the course also ties in the Artificial Intelligence (AI) program.

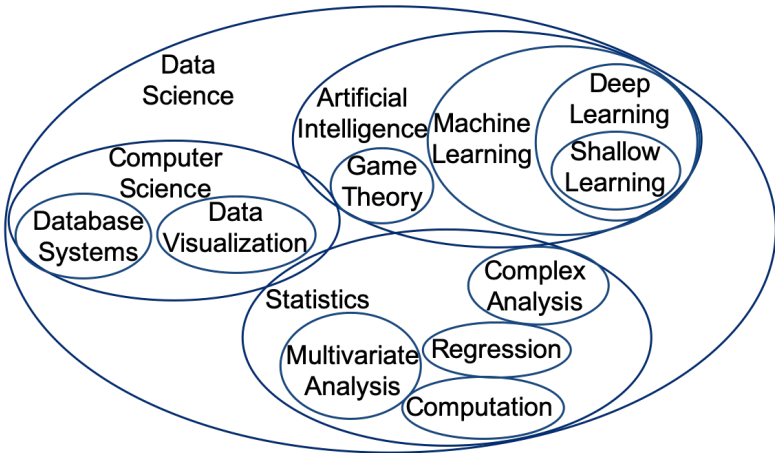

Figure 1: This figure demonstrates how various disciplines tie together for the fields of artificial intelligence, computer science, data science, and statistics.

Contents

1 Data Science

The Data Science curriculum focuses on the fundamentals of computer science, statistics, and applied mathematics, while incorporating real-world examples. By learning from practicing engineers and data scientists, graduates are prepared to succeed in specialized jobs involving everything from the data pipeline and storage to statistical analysis and eliciting the story the data tells..

2 685.621 Algorithms for Data Science

The prerequisites are important when a Computer Science degree was not attained as an undergraduate. The foundations learned in the EN.605.202 Data Structures course are a key component when working with data in any programming language. The understanding of EN605.201 Introduction to Programing using Java is important for having the necessary background in developing code that will utilize the algorithms designed in the Algorithms for Data Science course. Discrete Mathematics allows an understanding of ties between mathematic concepts and their relation to Computer Science.

The course description as listed Online with new modification is as follows:

The Algorithms for Data science course is a follow-on course to data structures (e.g., 605.202 Data Structures) providing a survey of computer algorithms, examining fundamental techniques in algorithm design and analysis, while developing skills in problem-solving which are required in all programs of study involving data science. Topics include advanced data structures for data science (analyzing algorithms, designing algorithms, asymptotic notation), algorithm analysis and computational complexity (recurrence relations, big-O notation, introduction to complexity classes (P, NP and NP-completeness)), data transformations (principal component analysis), design paradigms (divide and conquer, greedy heuristic, dynamic programming), and graph algorithms (depth-first and breadth-first search, ordered and unordered trees). Advanced topics are selected from among the following: approximation algorithms, data analysis, data probabilities and distributions, data processing techniques, game theory, linear programming, machine learning, matrix operations, optimization, and statistical learning methods. The course will draw on applications from Data Science.

3 Course Outline

This course outline is provided in more detail in a separate document with specifics of each module. Each course module runs for a period of seven (7) days, referred to as one week. The following is a summary of what is covered in the course for each of the 14 modules.

1. Data Structures - Analyzing Algorithms, Designing Algorithms, Asymptotic Notation
2. Advanced Data Structures - Binary Search Tree, Tree Traversal, Querying a Binary Search Tree, Searching, Minimum, Maximum, Successor, Predecessor, Insertion and Deletion
3. Basic Algorithm Analysis - Analyzing Algorithms, Designing Algorithms, Asymptotic Notation, Substitution Method for Solving Recurrences, Recursion Tree Method for Solving Recurrences, Master Method for Solving Recurrences
4. Probabilities and Distributions - Probabilities, Distributions, Covariance Matrix, Random Numbers, Hypothesis Testing
5. Data Processing - Data Types, Data Mining, Data Preparation, Outlier Removal, Data Cleansing, Feature Ranking, Feature Selection, Dimensionality Reduction
6. Data Transformations - Huffman Codes, Principal Component Analysis, Wavelets, Discrete Fourier Transform, Discrete Cosine Transform, Generating Features, Dimensionality Reduction

7. Computational Statistics - Expectation Maximization, Mixture Models, Bayes Classifier
8. Artificial Intelligence - What is AI, Rational Agents, AI Prehistory, History of AI, State of the Art, Agents and Environments, Rationality
9. Machine Learning - Linear Discriminant Analysis, Bayes Classifier, k-Nearest Neighbors, Mahalanobis Distance, Kernel Fisher's Discriminant, Parzen Window, Radial Basis Function Neural Networks, Probabilistic Neural Networks, Support Vector Machine
10. Game Theory - Introduction to Two-Player Games, Basic Concepts, Agents, Environments, Search Strategies, Search Methods, Making a Smart Agent.
11. Graph Algorithms - Breadth-First Search, Depth-First Search, Strongly Connected Components, Growing a Minimum Spanning Tree, The Algorithms of Kruskal and Prim.
12. Deep Learning - Feed Forward Neural Networks, Convolutional Neural Networks
13. Optimization - Matrix Chain Multiplication, Elements of Dynamic Programming, Optimal Binary Search Trees, Elements of the Greedy Strategy, Huffman Codes, Standard and Slack Forms, Formulating Problems as Linear Programs, Simplex Algorithm, Duality
14. NP-Complete - Turing Machines, Polynomial-Time Verification, NP-Completeness and Reducibility, NP-Completeness Proofs, The Hamiltonian-Cycle Problem, The Traveling-Salesman Problem

4 Assignments

There are 7 assignments due for the semester, 5 Home Works (HW) and 2 Programming Assignments (PA). The home works are worth 10% each and the programming assignments are worth 15% each for the the overall semester grade making the total of the assignments worth 80% of your overall grade.

5 Collaboration

Active student participation is an essential part of any online course. Therefore, part of your grade (00%) will be based on class participation. It consists of the following components.

- **Weekly module discussion participation** - 10%
- **Homework collaborative group participation** - 10%

6 Data Science Core Courses

The following list of prerequisites and required courses outline the theoretical and practical components of the Data Science curriculum. These courses separate the differences in Data Analysis and Data Science.

PREREQUISITE COURSES
605.101 Introduction to Python
605.201 Introduction to Programming Using Java
605.202 Data Structures
605.203 Discrete Mathematics
625.201 General Applied Mathematics
625.250 Multivariable and Complex Analysis
625.251 Introduction to Ordinary and Partial Differential Equations

FOUNDATION COURSES
685.621 Algorithms for Data Science
625.603 Statistical Methods and Data Analysis

REQUIRED COURSES
685.648 Data Science
605.662 Data Visualization
625.661 Statistical Models and Regression

AND ONE OF THE FOLLOWING:
605.641 Principles of Database Systems
605.649 Introduction to Machine Learning

AND ONE OF THE FOLLOWING:
625.615 Introduction to Optimization
625.664 Computational Statistics

6.1 625.603 Statistical Methods and Data Analysis

This course introduces statistical methods that are widely used in modern applications. A balance is struck between the presentation of the mathematical foundations of concepts in probability and statistics and their appropriate use in a variety of practical contexts. Foundational topics of probability, such as **probability rules, related inequalities, random variables, probability distributions, moments, and jointly distributed random variables**, are followed by foundations of statistical inference, including estimation approaches and properties, **hypothesis testing**, and model building. Data analysis ranging from descriptive statistics to the implementation of common procedures for estimation, hypothesis testing, and model building is the focus after the foundational methodology has been covered. Software, for example R-Studio, will be leveraged to illustrate concepts through simulation and to serve as a platform for data analysis.

6.2 685.648 Data Science

This course will cover the core concepts and skills in the emerging field of data science. These include problem identification and communication, probability, statistical inference, visualization, extract/transform/load (ETL), exploratory data analysis (EDA), linear and logistic regression, model evaluation and **various machine learning algorithms such as random forests, k-means clustering, and association rules**. The course recognizes that although data science uses machine learning techniques, it is not synonymous with machine learning. The course emphasizes an understanding of both data (through the use of systems theory, probability, and simulation) and algorithms (through the use of synthetic and real data sets). The guiding principles throughout are communication and reproducibility. The course is geared towards giving students direct experience in solving the programming and analytical challenges associated with data science.

6.3 605.662 Data Visualization

This course explores the underlying theory and practical concepts in creating visual representations of large amounts of data. It covers the core topics in data visualization: **data representation, visualization toolkits, scientific visualization**, medical visualization, information visualization, flow visualization, and volume rendering techniques. The related topics of applied human perception and advanced display devices are also introduced.

6.4 625.661 Statistical Models and Regression

Introduction to regression and linear models including least squares estimation, maximum likelihood estimation, the Gauss-Markov Theorem, and the Fundamental Theorem of Least Squares. Topics include **estimation, hypothesis testing**, simultaneous inference, model diagnostics, **transformations**, multicollinearity, influence, model building, and **variable selection**. Advanced topics include nonlinear regression, robust regression, and generalized linear models including logistic and Poisson regression.

6.5 605.641 Principles of Database Systems

This course examines the underlying concepts and theory of database management systems. Topics include database system architectures, data models, query languages, conceptual and logical database design, physical organization, and transaction management. The entity-relationship model and relational model are investigated in detail, object-oriented databases are introduced, and legacy systems based on the network and hierarchical models are briefly described. Mappings from the conceptual level to the logical level, integrity constraints, dependencies, and normalization are studied as a basis for formal design. Theoretical languages such as the relational algebra and the relational calculus are described, and high-level languages such as SQL and QBE are discussed. An overview of file organization and access methods is provided as a basis for discussion of heuristic query optimization techniques. Finally, transaction processing techniques are presented with a specific emphasis on concurrency control and database recovery.

6.6 605.649 Introduction to Machine Learning

Analyzing large data sets ("Big Data"), is an increasingly important skill set. One of the disciplines being relied upon for such analysis is machine learning. In this course, we will approach machine learning from a practitioner's perspective. We will examine the issues that impact our ability to learn good models (e.g., the curse of dimensionality, the bias-variance dilemma, and no free lunch). We will then examine a variety of approaches to learning models, covering the spectrum from **unsupervised to supervised learning**, as well as **parametric versus non-parametric methods**. Students will explore and implement several learning methods, including logistic regression, **Bayesian classification, decision trees, and feed-forward neural networks**, and will incorporate strategies for addressing the issues impacting performance (e.g., regularization, clustering, and dimensionality reduction). In addition, students will engage in online discussions, focusing on the key questions in developing learning systems. At the end of this course, students will be able to implement and apply a variety of machine learning methods to real-world problems, as well as be able to assess the performance of these algorithms on different types of data sets.

6.7 625.615 Introduction to Optimization

This course introduces applications and **algorithms for linear optimization, network optimization**, integer optimization, and nonlinear optimization. Topics include the primal and dual simplex methods, network flow algorithms, branch and bound, interior point methods, Newton and quasi-Newton methods, and heuristic methods. Students will gain experience in formulating models and implementing algorithms using MATLAB. No previous experience with the software is required.

6.8 625.664 Computational Statistics

Computational statistics is a branch of mathematical sciences concerned with efficient methods for obtaining numerical solutions to statistically formulated problems. This course will introduce students to a variety of computationally intensive statistical techniques and the role of computation as a tool of discovery. Topics include **numerical optimization in statistical inference [expectation-maximization (EM) algorithm, Fisher scoring, etc.], random number generation**, Monte Carlo methods, randomization methods, jackknife methods, bootstrap methods, tools for identification of structure in data, estimation of functions (orthogonal polynomials, splines, etc.), and graphical methods. Additional topics may vary. Coursework will include computer assignments.

6.9 625.603 Statistical Methods and Data Analysis

This course introduces statistical methods that are widely used in modern applications. A balance is struck between the presentation of the mathematical foundations of concepts in probability and statistics and their appropriate use in a variety of practical contexts. Foundational topics of probability, such as **probability rules, related inequalities, random variables, probability distributions, moments, and jointly distributed random variables**, are followed by foundations of statistical inference, including estimation approaches and properties, **hypothesis testing**, and model building. Data analysis ranging from descriptive statistics to the implementation of common procedures for estimation, hypothesis testing, and model building is the focus after the foundational methodology has been covered. Software, for example R-Studio, will be leveraged to illustrate concepts through simulation and to serve as a platform for data analysis.

6.10 685.648 Data Science

This course will cover the core concepts and skills in the emerging field of data science. These include problem identification and communication, probability, statistical inference, visualization, extract/transform/load (ETL), exploratory data analysis (EDA), linear and logistic regression, model evaluation and **various machine learning algorithms such as random forests, k-means clustering, and association rules**. The course recognizes that although data science uses machine learning techniques, it is not synonymous with machine learning. The course emphasizes an understanding of both data (through the use of systems theory, probability, and simulation) and algorithms (through the use of synthetic and real data sets). The guiding principles throughout are communication and reproducibility. The course is geared towards giving students direct experience in solving the programming and analytical challenges associated with data science.

7 Artificial Intelligence Core Courses

The following list of prerequisites and required courses outline the theoretical and practical components of the Artificial Intelligence curriculum. The part-time Artificial Intelligence program will educate and train practicing scientists and engineers to be able to carry out engineering and scientifically oriented research and development using their artificial intelligence knowledge and skills.

The rigorous curriculum will provide engineers and computer scientists with a working knowledge of the theoretical concepts in artificial intelligence and will also provide the students with the knowledge and skills to apply both current and future theoretical concepts to real systems and processes. The course content will be based on the foundational content embodied in the current computer science courses modified to provide relevant examples in the artificial intelligence setting.

Courses are offered online as well as in-person at the Applied Physics Laboratory.

John A. Piorkowski, Program Chair
Principal Professional Staff
JHU Applied Physics Laboratory

8 AI Program Requirements

In order to earn a Master of Science in Artificial Intelligence, the student must complete 30 approved credits within five years. The curriculum consists of 12 credits of core courses and 18 or more credits of electives from the Artificial Intelligence program. Nine (9) credits must be taken at the 700-level. One or more core courses can be waived by the student's advisor if a student has received an A or B in equivalent graduate courses. In this case, the student may replace the waived core courses with the same number of other graduate Artificial Intelligence courses and may take these courses after all remaining core course requirements have been satisfied. Only one C-range grade (C+ C, C-) can count toward the master's degree. All course selections are subject to advisor approval.

AI CORE FOUNDATION COURSES

685.621 - Algorithms for Data Science
705.601 - Applied Machine Learning (For an applied approach)
605.645 - Artificial Intelligence
705.603 - Creating AI-Enabled Systems

All students must take at least 6 of the following courses:

525.661 - UAV Systems and Control
525.670 - Machine Learning for Signal Processing
525.724 - Introduction to Pattern Recognition
525.733 - Deep Learning for Computer Vision
525.770 - Intelligent Algorithms
525.786 - Human Robotics Interaction
605.613 - Introduction to Robotics
605.617 - Introduction to GPU Programming
605.624 - Logic: Systems, Semantics, and Models
605.635 - Cloud Computing
605.646 - Natural Language Processing
605.647 - Neural Networks
605.662 - Data Visualization
605.745 - Reasoning Under Uncertainty
605.746 - Advanced Machine Learning
605.747 - Evolutionary Computation
625.638 - Neural Networks
645.651 - Integrating Humans and Technology
695.637 - Introduction to Assured AI and Autonomy

8.1 705.601 - Applied Machine Learning

Machine Learning (ML) is the art of solving a computation problem using a computer without an explicit program. ML is now so pervasive that various ML applications such as image recognition, stock trading, email spam detection, product recommendation, medical diagnosis, predictive maintenance,

cybersecurity, etc. are constantly used by organizations around us, sometimes without our awareness. In this course, we will rigorously apply machine learning techniques to real-world data to solve real-world problems. We will briefly study the underlying principles of **diverse machine learning approaches such as anomaly detection, ensemble learning, deep learning with a neural network, etc.** The main focus will be applying tool libraries from the Python-based Anaconda and Java-based Weka data science platforms to datasets from online resources such as Kaggle, UCI KDD, open source repositories, etc. We will also use Jupyter notebooks to present and demonstrate several machine learning pipelines.

8.2 605.645 - Artificial Intelligence

The incorporation of advanced techniques in reasoning and problem solving into modern, complex systems has become pervasive. Often, these techniques fall within the realm of artificial intelligence. This course focuses on **artificial intelligence from an agent perspective and explores issues of knowledge representation and reasoning.** Students will investigate a wide variety of approaches to artificial intelligence including **heuristic search** and stochastic search, logical and probabilistic reasoning, planning, learning, and perception. Advanced topics will be selected from areas such as robotics, vision, natural language processing, and philosophy of mind. Students will have the opportunity to explore both the philosophical and practical issues of artificial intelligence during the course of the class.

8.3 705.603 - Creating AI-Enabled Systems

Achieving the full capability of AI requires a system perspective to effectively leverage algorithms, data, and computing power. Creating AI-enabled systems includes thoughtful consideration of an operational decomposition for AI solutions, **engineering data for algorithm development**, and deployment strategies. To realize the impact of AI technologies requires a systems perspective that goes beyond the algorithms. The objective of this course is to bring a system perspective to creating AI-enabled systems. The course will explore the full-lifecycle of creating AI-enabled systems starting with problem decomposition and addressing data, design, diagnostic, and deployment phases. The course will also cover ethics and bias in AI systems. The course includes a systems project that will encompass the full-lifecycle with interim milestones throughout the course. Homework assignments will be provided that in involves python programming.

8.4 525.724 - Introduction to Pattern Recognition

This course focuses on the underlying principles of pattern recognition and on the methods of machine intelligence used to develop and deploy pattern recognition applications in the real world. Emphasis is placed on the pattern recognition application development process, which includes problem identification, concept development, algorithm selection, system integration, and test and validation. **Machine intelligence algorithms to be presented include feature extraction and selection, parametric and non-parametric pattern detection and classification, clustering, artificial neural networks, support vector machines**, rule-based algorithms, fuzzy logic, genetic algorithms, and others. Case studies drawn from actual machine intelligence applications will be used to illustrate how methods such as pattern detection and classification, signal taxonomy, machine vision, anomaly detection, data mining, and data fusion are applied in realistic problem environments. Students will use the MATLAB programming language and the data from these case studies to build and test their own prototype solutions.

8.5 605.647 - Neural Networks and 625.638 - Neural Networks

This course provides an introduction to concepts in neural networks and connectionist models. Topics include parallel distributed processing, **learning algorithms, and applications.** Specific networks discussed include Hopfield networks, bidirectional associative memories, perceptrons, **feedforward networks with back propagation**, and competitive learning networks, including self-organizing and Grossberg networks. Software for some networks is provided. Prerequisite(s): Multivariate calculus and linear algebra. Course Note(s): This course is the same as 625.638 Neural Networks.

References

[1] Christopher Bishop, *Pattern Recognition and Machine Learning*, Springer, 2006, https://www.microsoft.com/en-us/research/uploads/prod/2006/01/Bishop-Pattern-Recognition-and-Machine-Learning-2006.pdf

[2] Thomas H. Cormen, Charles E. Leiserson, Ronal L. Rivest, and Clifford Stein, *Introduction to Algorithms*, 3rd Edition, MIT Press, 2009

[3] Ian Goodfellow, Yoshua Bengio and Aaron Courville, *Deep Learning*, MIT Press, 2016, https://www.deeplearningbook.org/

[4] Stuart Russell and Peter Norvig, *Artificial Intelligence a Modern Approach* Fourth Edition, Pearson, 2020