INTRODUCTION TO MACHINE LEARNING

INTRODUCTION TO THE COURSE
Self Introduction

- Mingon Kang, PhD
  - Homepage: http://mkang.faculty.unlv.edu/

- Research interests:
  - Bioinformatics, Machine Learning, Data Mining, and Big Data Analytics

- Projects you may be interested in:
  - Biomedical projects
  - Computer Vision projects
  - Developing Python package
Course Information

- Instructor: Dr. Mingon Kang
- Office: SEB 3214
- Email: mingon.kang@unlv.edu
  - Only reply to e-mails that are sent from UNLV student email accounts and list the course number

- Office Hours:
  - Tu/Th, 1-2:30pm
  - By appointment

- Course Materials
  - Homework assignments, lecture slides, and other materials will be posted in the webpage.
  - All lectures may be recorded (but no guaranteed)
Course Description

This course covers various machine learning algorithms for regression, classification, clustering, and ensemble learning. Students will learn applying machine learning techniques to solve challenging problems in various fields.
Topics

- Data representation and visualization
- Supervised Learning
  - K-Nearest Neighbor (KNN)
  - Linear Models
  - Logistic Regression
  - Neural Networks
  - Convolutional Neural Networks
  - Support Vector Machine (SVM) and Kernels
  - Markov Model and Hidden Markov Model
Topics

- Unsupervised Learning
  - Hierarchical Clustering
  - Image segmentation
  - Restricted Boltzmann Machine
  - Deep Belief Network
- Ensemble Learning
Choice of Language

- There are many programming languages such as C/C++, JAVA.
- However, high-level script languages such as R, MATLAB, Python are highly recommended.
- Why?
  - Better for file I/O of textual data
  - Better to do matrix manipulation
  - Fast Prototyping
- We will use Python for this course.
Reference

- Deep Learning, Vol. 1: From Basics to Practice by Andrew Glassner, 2018
Reference

  - Webpage: http://www.deeplearningbook.org
Reference

- Pattern Recognition and Machine Learning, Christopher M. Bishop, 6-edition, Springer-Verlag New York, 2006
Evaluation (tentative)

- Attendance (5%)
  - If a student misses more than 4 sessions (class meetings), the student's final grade for the course may be reduced by 5%.

- Homework Assignment (30%)
  - All programming assignments

- Exams (50%)
  - Exam 1 (25%) and Exam 2 (25%)

- Project (12%)

- Presentation (3%)
  - Graduate students will be asked additional presentations.

- Poster (up to additional 5%)
  - Mini-workshop for poster presentation

- Late submission policy:
  - Late assignments will NOT be accepted for credit.
Homework Assignment

- For every homework assignments, include the following files:
  - A word file: Homework description. Should include all data, figure, and tables required.
    - Specify what python version used and what packages are used.
  - Source files:
    - *.py: Do not submit python notebook file. All description must be described in the word file. The “py” file will be tested in command. Should specify python version in the header.
    - Change the extension name from “py” to “.py.txt”.

- If fail to submit all of the files, grade will be zero.
- If submitted wrong files, grade will be zero.
- It is your responsibility to check whether your submission is all correct.
## Grade Evaluation

<table>
<thead>
<tr>
<th>Grade</th>
<th>Percentage Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>[90% - 100%]</td>
</tr>
<tr>
<td>A-</td>
<td>[85% - 90%]</td>
</tr>
<tr>
<td>B+</td>
<td>[80% - 85%]</td>
</tr>
<tr>
<td>B</td>
<td>[75% - 80%]</td>
</tr>
<tr>
<td>B-</td>
<td>[70% - 75%]</td>
</tr>
<tr>
<td>C+</td>
<td>[65% - 70%]</td>
</tr>
<tr>
<td>C</td>
<td>[60% - 65%]</td>
</tr>
<tr>
<td>C-</td>
<td>[55% - 60%]</td>
</tr>
<tr>
<td>D</td>
<td>[45% - 55%]</td>
</tr>
<tr>
<td>F</td>
<td>[0% - 45]</td>
</tr>
</tbody>
</table>
Academic Integrity

- Academic dishonesty
  - Cheating
  - Plagiarism
  - Collusion
- The submission for credit of any work or materials that are attributable in whole or in part to another person
- Taking an examination for another person
- Any act designed to give unfair advantage to a student or the attempt to commit
Before beginning the course

- Let’s discuss about the origins of Computer Science
Philosophy

- **Definition of the word**
  - “The study of the fundamental nature of knowledge, reality, and existence, especially when considered as an academic discipline.” Oxford Dictionary

- **Literally means “love of wisdom” or “friend of wisdom”**

- **Logic**
  - logically describe world (around 500 BC)

- **Ancient Graeco-Roman philosophy**
  - Socrates, Plato, Aristotle, and etc.
Philosophers

- Aristotle
- Gottfried Wilhelm Leibniz
- George Boole
- Bertrand Russell
- Alan Turing
Aristotle (384 – 322 BC)

- So many different roles
  - Physics, Biology, Music, Linguistics, Zoology, Economy, Politics
- How to understand the different world?
  - LOGIC
Gottfried Wilhelm Leibniz

- German philosopher (1646-1716)
- Known as one of the founding fathers of calculus
- Wanted to prove all phenomena using binary logic
  - Convert world to binary logic
George Boole

- English mathematician, philosopher, and logician (1815-1864)
- Author of “The Laws of Thought”
- Inventor of Boolean Logic

Note that Boolean logic can be used to implement binary arithmetic
Bertrand Russell

- British philosopher, logician, mathematician, historian, writer, social critic and political activist
- Wanted to make perfect mathematics from perfect logic
- Author of “Principia Mathematica”, published in 1910, 1912, and 1913.

Total of 1994 pages!!
*54.43. \[ \vdash \alpha, \beta \in 1 \vdash \alpha \cap \beta = \Lambda \equiv \alpha \cup \beta \in 2 \]

**Dem.**

\[ \vdash *54.26. \vdash \alpha = \iota x. \beta = \iota y. \vdash \alpha \cup \beta \in 2 \equiv \iota x \equiv y. \]

\[ [*51.231] \]

\[ [*13.12] \]

\[ \vdash \iota (1). *11.11.35. \vdash \vdash \vdash \vdash \vdash \iota (\forall x, y). \alpha = \iota x. \beta = \iota y. \vdash \alpha \cup \beta \in 2 \equiv \alpha \cap \beta = \Lambda \]

\[ (2) \]

From this proposition it will follow, when arithmetical addition has been defined, that \(1 + 1 = 2\).
Alan Turing (1912-1954)

- Automatize logic.
  - If everything can be explained by logic, we may implement the logic automatically not manually.

- Turing Machine
  - A model of a general purpose computer
Summary

Aristotle (384-322BC) modern disciplines

Gottfried Wilhelm Leibniz (1646-1716) binary logic

George Boole (1815-1864) Boolean Logic

Bertrand Russell (1872-1970) Principia Mathematica

Alan Turing (1912-1954) Automated logic

See http://www.datesandevents.org/events-timelines/07-computer-history-timeline.htm