Lecture 1 EE 559 Jan 8, 2024

Machine Learning I: Supervised Methods

B. Keith Jenkins

Announcements

- The website for the class is at Desire2Learn (D2L):
  - courses.uscden.net
  - Post lecture notes, lecture videos, assignments, handouts
  - Turn in your assignments
  - View your grades
- Piazza will be used for:
  - Course announcements
  - Discusion/Q&A forum
  - You will get an invitation by email

Reading

- Bishop Sec. 1.0, 1.1

Today's Lecture

- Introduction to the course and to machine learning
- Course administrative information (syllabus)
- Basic concepts in supervised ML (part 1) } → Lecture 2

What is Machine Learning?

Examples of ML problems or applications:

- Image classification (e.g., object recognition)
- Curve fitting
- Natural language processing (NLP)
- Modelling of data (inputs → outputs)
- Image reconstructing (e.g., image enhancement – deblur or increase resolution).
- Recommender systems.
- Predictions / forecasting – e.g., stock market, weather forecasting

Examples of problems or applications solved by computer without ML:

- Chemical processes that can be modelled by differential equations
- Computing Fast Fourier Transform
- Path finding
- Network security
- Decoding, coding, compression

Characteristics of problems best solved using ML:

- Availability and use of data
- Existence of (learnable) transformation from input to output
- → Transformation from inputs to outputs is unknown.
- Not solvable with a small number of rules in a rule-based (expert) system.
- Can be posed and solved as an optimization problem.
- Known models (e.g., based on physics) take too long to compute.

Characteristics of problems best solved by computer without ML:

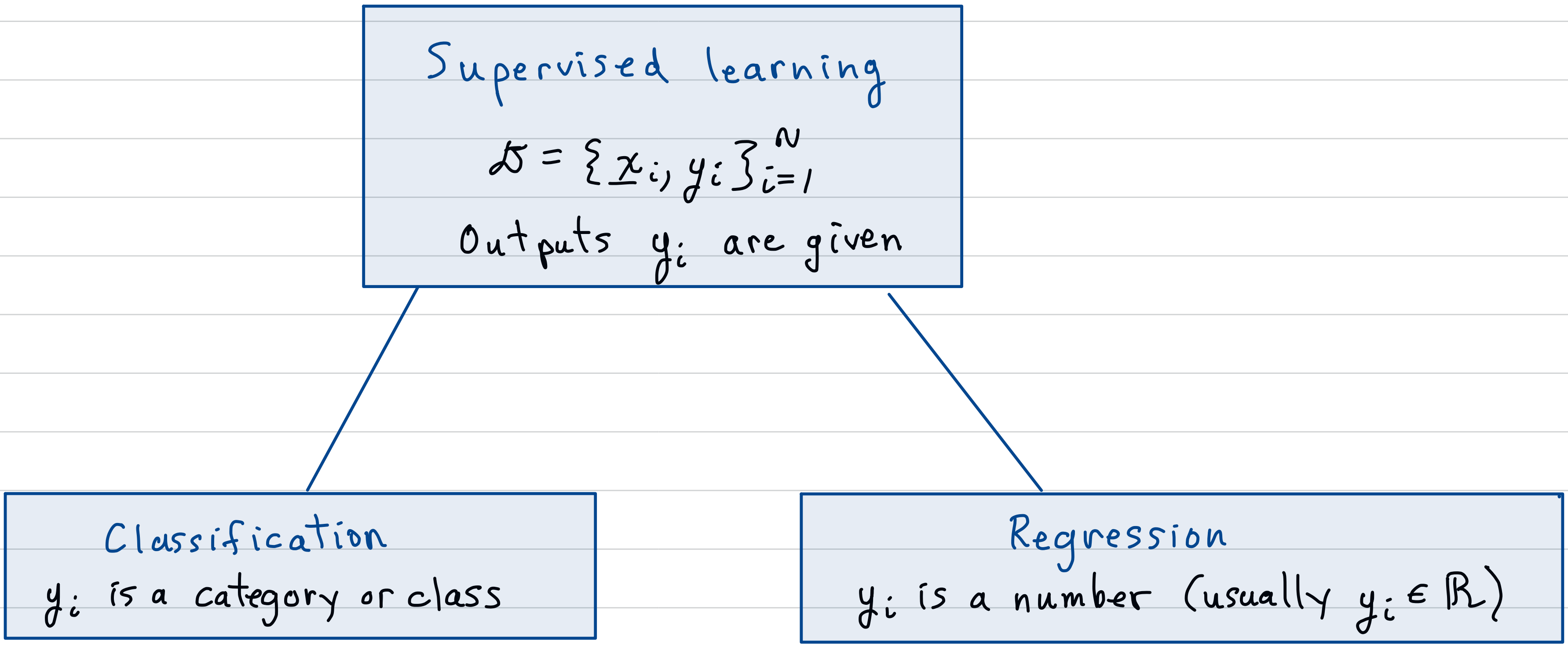
- Transformation from inputs to outputs is known and calculable.
- Not enough data to learn from.
- Problem can be solved with or without ML, but without-ML solution is deemed better.
- Requires trust and explainable results, and the ML approach uses models that aren't explainable.

What about the following?

	Use ML	Use non-ML techniques
Recommend movies	Yes	Can, but less effective
Weather prediction - 1960's	1-neuron system demonstrated vs.	Mathematical/numerical models under development
Weather prediction - now	Starting to be used	Mostly (numerical) comp. based on equations)
Weather prediction - future	Likely more ML (help with human inference)	Also numerical comp. based on equations
Face recognition	Yes	No
Arithmetic on integers	Can be done! (not very efficient)	Yes (very efficient)
Spaceship autopilot	Yes (e.g., finding good landing site on another planet; optimizing fuel use in real time)	Yes (e.g., planning orbital trajectories)

ML algorithms learn from <u>data</u> $\mathcal{D} = \{ \underline{x}_i, y_i \}_{i=1}^{N}$, $N = \#$ data points.

inputs (e.g., pixel values of a picture) outputs: (e.g., dog or cat)



Supervised learning

$\mathcal{D} = \{ \underline{x}_i, y_i \}_{i=1}^{N}$

Outputs y_i are given

Classification

y_i is a category or class

Regression

y_i is a number (usually $y_i \in \mathbb{R}$)

Ex: $y_i \in \{\text{dog}, \text{cat}\}$ $\forall i$.

Ex: y_i = selling price of a house

→ All covered in EE 559

Other modes for ML

EE660

Unsupervised learning

No outputs y_i. $\mathcal{D} = \{\underline{x}_i\}_{i=1}^{N}$.

Ex: $\underline{x}_i$ = pixel values of pictures of possible Mars landing sites.

Semi-supervised learning

Some labeled + some unlabeled data.

Ex: some MRI pictures, labeled by experts + lots of unlabeled MRI pictures.

EE556

Reinforcement learning

Inputs are environment (states); outputs are its actions. Feedback provided from reward after (many) actions.

Ex: learn to play a game; learn to move a robot arm.

Comment

EE541, EE641 focus on deep (neural-network) learning, and include deep-learning approaches to:

Supervised learning (classif. and regression)
Some unsupervised learning
Some reinforcement learning

p.7

USC
Viterbi
School of Engineering

EE 559
Machine Learning I: Supervised Methods
Spring 2024
Units: 4

Lecture: MW 4:00 – 5:50 PM Pacific Time
Discussion: Fri 11:00 – 11:50 AM Pacific Time
Location: OHE 122 and DEN@Viterbi
Course URL: https://courses.uscden.net/d2l/home

Instructor: B. Keith Jenkins
Office: EEB 404A
Email: bjenkins@usc.edu [please include "EE 559" in the subject line]
Office Hours and location: MW 11:00 AM - 12:30 PM in EEB 403 and zoom
Zoom link (office hours): See D2L calendar

Teaching Assistants:

TA: Shipeng Liu (25% TA)]
Email: shipengl@usc.edu [please include "EE 559" in the subject line]
Office Hours: TBD

TA: Sagar Sudhakara (50% TA)]
Email: sagarsud@usc.edu [please include "EE 559" in the subject line]
Office Hours: TBD

TA: Jintang Xue (25% TA)]
Email: jintangx@usc.edu [please include "EE 559" in the subject line]
Office Hours: TBD

Graders: TBA

IT Help: For help with coding machine-learning algorithms, consult piazza, online forums, or the TAs; for help with other python coding, working with datasets, or using library routines in python, also try online help and documentation, and reference resources given below; for help with USC-supplied software or on-campus networking, consult USC ITS at https://itservices.usc.edu/contact/ .

den tech support (videos, etc.):
dentsc@usc.edu

Course Description (Catalogue)

Distribution-free and probabilistic methods for supervised classification and regression; learning algorithms; optimization techniques; feature-space transformations; parametric and nonparametric methods; Bayes decision theory; artificial neural networks.

Course Description (Expanded)

Concepts and algorithms for pattern recognition and regression using machine learning are covered in depth. The course will stress an understanding of different supervised-learning algorithms at both theoretical and practical levels, as well as their advantages and disadvantages. Underlying fundamentals are emphasized, including theory and origins of learning algorithms and criterion functions. The goal is to give the student an understanding of some fundamental approaches to machine learning, to enable further study and growth on their own. The student's work will include mathematical analysis, analytical understanding, and writing and running code that learns from data. A moderately sized project in the second half of the semester will involve developing and optimizing one or more machine learning systems to perform well on real-world datasets. This course is intended for graduate students in Electrical and Computer Engineering or related fields, who wish to gain an understanding of, and some experience with, machine learning approaches, tools, and techniques.

Learning Objectives

After successfully completing this course, the student will:

- Have a perspective of different approaches to supervised machine learning for pattern classification and regression
- Understand the underlying math of a variety of supervised-learning methods
- Be able to use statistical and non-statistical techniques to solve machine learning problems
- Be able to code and run algorithms for learning from data
- Be able to optimize machine learning algorithms and systems, and assess their overall performance
- Know how to create new techniques for machine learning where needed

Co-Requisites: EE 503 and EE 510

Recommended Preparation: knowledge of Python (e.g., at the level of EE 541), knowledge of multivariate calculus (at the sophomore or junior undergraduate level). For students that don't know Python, resources will be provided to help them learn Python during the first few weeks of the semester, leveraging from their knowledge of MATLAB; students should allow some extra time for this if they don't know much Python.

Course Notes

- This course will use D2L for our course materials. This will include (accessible to all EE 660 students):
  - Course lecture notes (posted before and after each lecture)
  - Assignments, homework solutions, etc.
  - Videos of lectures and discussion sessions posted after each event.
  - A place to upload your completed assignments and project reports
  - A place to view your graded materials and course gradebook
- This course will use piazza for its online discussion forum:
  - All course announcements
  - Questions and answers relating to hoemworks, lectures, etc.
  - All students will get an invitation to the ~~EE 660 Fall 2024~~ EE 559 Spring 2024 piazza website. Be sure to check that you have access to receive course announcements and participate in the Q & A.

Technological Proficiency and Hardware/Software Required

Python will be used throughout this class for homework assignments and the class project. All students will be responsible for installing and maintaining their own Python distribution.

For help resources, see "IT Help" on the first page of this syllabus above, and "Supplementary resources for Python" on the next page below.

Required Readings and Supplementary Materials (not yet final)

Required text

- C. M. Bishop, *Pattern Recognition and Machine Learning* (Springer, 2006). ISBN-13: 978-0387-31073-2. Available from USC bookstore, Amazon.com, and Springer.com.

Supplementary texts

- Ali H. Sayed, *Inference and Learning from Data, Vol.3: Learning* (Cambridge University Press, 2022)
- Kevin Murphy, *Probabilistic Machine Learning: An Introduction* (MIT Press, 2022). Preprint available for download at https://probml.github.io/pml-book/book1.html
- Ali H. Sayed, *Inference and Learning from Data, Vol.1: Foundations* (Cambridge University Press, 2022)
- R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*, Second Edition (Wiley-Interscience, John Wiley and Sons, Inc., New York, 2001)
- T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, Second Edition (Springer, 2009)
- I. Goodfellow, Y. Bengio, and A. Clourville, *Deep Learning* (MIT Press, 2016)
- Ethem Alpaydin, *Introduction to Machine Learning, Fourth Edition* (MIT Press, 2020)
- Jeremy Watt, Reza Borhani, Aggelos Katsaggelos, *Machine Learning Refined*, 2nd Ed., (Cambridge University Press, 2020).

Supplementary resources for Python

- Hans Fanghor, *Introduction to Python for Computational Science and Engineering* (2016) [PCSE], available for free download at: https://github.com/fangohr/introduction-to-python-for-computational-science-and-engineering

- Fabio Nelli, *Python Data Analytics*, (APress, 2015) [PDA], available for download from USC Library: http://usc.summon.serialssolutions.com/search?q=Fabio%20Nelli%2C%20Python%20Data%20Analytics#!/search?ho=t&l=en&q=Fabio%20Nelli,%20Python%20Data%20Analytics

- The Python 3 Tutorial [PT]: https://docs.python.org/3.8/tutorial/index.html

- EU Python 3 Tutorial [EUP]: Good for Chapters on object oriented programming, class vs. instance attributes, and inheritance http://www.python-course.eu/python3_course.php

- NumPy for Matlab Users:
  - https://mathesaurus.sourceforge.net/matlab-numpy.html
  - https://numpy.org/doc/stable/user/numpy-for-matlab-users.html

- Robert Johansson, *Numerical Python*, http://jrjohansson.github.io/numericalpython.html

- CS61a from UC Berkeley (full class on Python).

p.10

Course outline– will be discussed in Lecture 2

Description and Assessment of Assignments

1. Homework assignments

Homework assignments will, on average, consist of approximately 50% computer problems and 50% analytical problems. Homework assignments will be posted on the course website when assigned, and typically due 1 week later (due date and time will be specified on the homework assignment).

Python 3 is required for all homework computer problems, with the exception of the first 2 homework assignments, for which you are also allowed to use MATLAB. Some of the computer problems will require coding machine learning algorithms yourself (without using libraries such as pandas, sk-learn, etc.); others will allow use of these and other libraries. Each assignment will state the requirements and guidelines for its computer problems.

Python 3 is open source and available for free download (*e.g.*, as a distribution like www.anaconda.com).

2. Midterm and final exams

On-campus students and den students local to campus will be required to take the exams in person, except for documented medical or emergency reasons. The midterm exam will either be a ~5-day take-home midterm assignment, or an in-class midterm exam. It will be scheduled to take place during Week 9 (3/4-3/8), and if in-class, it will be Wed., 3/6/2023, 4:00 – 5:50 PM, Pacific Standard Time).

The final exam is scheduled by the university and will be Wednesday, May 1, 4:30 – 6:30 PM, Pacific Daylight Time.

3. Course project

Programming languages: For the course project, you will use python 3. You may also optionally use C/C++ for portions of your code where necessary (e.g., for faster compute time); for this, or to use any other language for portions of your code, check with the instructor or a TA first.

Typical project content (exact content to be determined): You will be given a choice of 2-3 real-world datasets and associated nontrivial problems to solve. Your goal is to design and demonstrate a good-performing machine learning system, to compare some different approaches, and to understand and explain the results you obtain. To do so, you will use your choice of preprocessing, feature extraction, dimensionality adjustment, and classifiers/regressors, etc., to solve the problem; as well as cross validation, training and test datasets as appropriate. You will be given some guidelines and suggestions, as well as some requirements of items your project must include.

Deliverables: On or before the due date specified in the Project Assignment, you will turn in your project by uploading 3 files to the D2L website: (1) a written final report (as a pdf file); (2) a single, computer-readable pdf file with all of your code; and (3) a zip file that contains all your code files (e.g., .py files) in a form that can be run to verify your results. The written final report must be typewritten and include tables and graphs to show your final results as well as sample or summary intermediate results. The suggested and maximum length will be given in the project assignment. Detailed instructions for the written final report will be posted along with templates in Word and LaTeX.

Sample project (from a previous semester): You are given a Hand Postures dataset that consists of frames taken from videos of different hand gestures. The acquired data consists of locations of markers on a glove worn by each subject. The dataset is challenging because: (1) the markers are not labelled, so it is not known which data entry corresponds to which marker; (2) typically some of the markers are occluded in any given frame (image); and (3) the number of features varies from one data point (one frame) to another. Your goal is to design and demonstrate a pattern recognition system that can reliably recognize the hand posture, given input data from a separate test set. Your project is required to include a trial of at least 4 classifiers/learning algorithms (some of which are specified, some of which are up to you); feature dimensionality adjustment; any necessary preprocessing; proper dataset usage including cross validation and model selection; estimate of your final system's performance on unseen data; analysis, explanation, and demonstration that you understand intermediate and final results.

Participants: Each project can be done by a team of 2 students, or by an individual student. Guidelines will be provided in the Project Assignment as to as to which datasets and problems constitute a reasonable topic and workload for a team project and for an individual project.

Timeline:

- Week 10: Project Assignment, and Dataset Descriptions, posted on D2L
- Week 14: Final Report and code files due.

There are no progress reports or proposals required. However, the instructors and TAs are available for any discussions or questions that you would find helpful; and you are encouraged to seek advice or direction as you deem appropriate for your project.

Your course project will be graded by the following criteria (tentative):

- Technical approach (20%);
- Analysis, understanding, and interpretation (20%);

p.12

- Performance of your final system (15%);
- Difficulty of the problem (15%);
- Workload (15%);
- Quality of final report write-up (15%),

Course Grading Breakdown

Assessment Tool (assignments)	**% of Grade**
Homework	20
Course project	24
Midterm exam	24
Final exam	24
Class participation	8
TOTAL	100

Extra credit homework problems (throughout the semester) will be tallied separately from the above scores, and will determine students' grades for borderline cases (*i.e.*, students near a border between two letter grades).

Assignment Submission Policy

Submit your homework by uploading one pdf file of your solution, and one computer-readable pdf file of all your code (for assignments with computer problems), to the Assignment Dropbox on the D2L website. A computer-readable pdf can be generated by converting an editable text document to pdf; Jupyter notebooks can be converted to pdf (or first html) using nbconvert. Scanned documents and screen-shots are not computer readable.

Due date and time will be stated on each homework assignment. The late submission policy will be posted on the D2L web site, and will allow for a given number of free late days. Late penalties will be uniformly applied to everyone. Exceptions will only be granted for unusual or emergency situations (e.g., documented emergencies or medical conditions).

Grading Timeline

Graded assignments including comments will be available on the D2L website as soon as grading is completed and verified (typically ~2 weeks after the due date).

Additional Policies

Policy on Collaboration and Individual Work in this Class

Collaboration on techniques for solving homework assignments and computer problems is allowed, and can be helpful; however, each student is expected to work out, code, and write up his or her own solution. Use of other solutions to homework assignments or computer problems, from any source including other students, before the assignment is turned in, is not permitted.

For class projects, general collaboration to resolve issues, or to clarify technical material, is allowed. Use of internet as well as journal and conference literature is encouraged. However, each student (or team) does their own work and writes up their own report. The author(s) of the report are presenting themselves as having done the work described in the report. Any reported work, explanations, information, or code that is obtained from others must be cited as such; instructions for doing this will be given with the project assignment. Including such work in the report or code without citing it amounts to plagiarism.

Please also see below for additional policies that apply to all USC classes.