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: Tu, Th 1:30-2:30 PM in EEB 403 and zoom
Zoom link (office hours): See D2L calendar

Teaching Assistants: TBA

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/ .

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

- Course lecture notes, assignments, homework solutions, etc. will be posted on piazza, which will also be used for discussion and questions. There will be a link to the course piazza page on D2L.
- Videos of lectures and discussion sessions will be posted on D2L, and will be accessible to all students.
- You will submit your completed assignments and project reports by uploading pdf files to D2L.
- You can view your graded materials and course gradebook on D2L.

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 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/ .

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

- Kevin Murphy, *Probabilistic Machine Learning: An Introduction* (MIT Press, 2022). Preprint available for download at https://probml.github.io/pml-book/book1.html .
- 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)
- Simon Haykin, *Neural Networks and Learning Machines,* 3rd Edition, (Pearson, 2009)
- 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):

Course outline (summary version)

> Order of topics covered will be slightly different than the order presented below
> Abbreviations: ML: machine learning; ANN: artificial neural networks

1. Course introduction
   – Basic concepts, ML paradigm, fundamental assumptions
2. Preliminaries
   – Discriminant functions; methods for mutlclass classification
   – Computational complexity
   – Convex functions
   – Fundamental assumptions
   – The learning problem
   – Criterion (objective) functions
   – Approaches to optimization
3. Learning and optimization 1
   – Gradient descent methods
   – Perceptron learning and convergence proof
   – Mean-squared-error algorithms for regression and classification
4. Nonlinear approaches for regression and classification
   – Nonlinear transformation by basis functions
   – Nonlinear transformation by kernel functions
5. Complexity in machine learning
   – Degrees of freedom and constraints
   – Regularization
   – Introduction to VC dimension
6. Learning and optimization 2
   – Optimization with constraints: Lagrangian techniques
   – Kernels and kernel methods
   – Support vector machines for classification
   – Support vector machines for regression
7. Validation and error estimation
   – Dataset usage: training, validation, test sets
   – Model selection and cross validation
8. Artificial neural networks 1
   – Single layer and multiple layer feedforward networks
   – Interpretations and capabilities
   – Learning algorithms
9. Artificial neural networks 2
   – ANNs as universal function approximators (proof by construction)

- Example: radial basis function networks
- Degrees of freedom and complexity in ANN

10. Feature selection and dimensionality reduction
    - Principal Components Analysis
    - Other linear transformations for 2-class and multiclass problems
11. Bayes decision theory for classification
    - Theoretically optimal probabililstic classification (minimum error and minimum risk criteria)
12. Density estimation techniques for classification
    - Mathematical approach and convergence
    - Kernel density estimation and k-nearest neighbors estimation
    - Discriminative and generative approaches
13. Density estimation techniques for regression
    - Theoretically optimal probabilistic MSE regression
    - k-nearest neighbors regression
    - Computational complexity and speed-up techniques (regression and classification)
14. Parametric estimation techniques (as time permits)
    - For classification
    - For regression
15. Conclusion and summary
    - Relation of topics to each other
    - ML key elements in the design phase, learning phase, and prediction phase

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, during a regular lecture day and time that week (either Mon., 3/4/2024, 4:00 – 5:50 PM, or Wed., 3/6/2023, 4:00 – 5:50 PM, Pacific Standard Time), TBA.

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 may use python 3 and/or C/C++. If you want to use any other language for portions of your project, please check with an instructor or 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 10-11: Project Assignment, and Dataset Descriptions, posted on D2L
- Week 15: 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%);

- Performance of your final system (15%);
- Difficulty of the problem (10%);
- Workload (15%);
- Quality of final report write-up (15%),

Course Grading Breakdown (tentative)

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; 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.

Statement on Academic Conduct and Support Systems

Academic Conduct:

Plagiarism – presenting someone else's ideas as your own, either verbatim or recast in your own words – is a serious academic offense with serious consequences. Please familiarize yourself with the discussion of plagiarism in SCampus in Part B, Section 11, "Behavior Violating University Standards" policy.usc.edu/scampus-part-b. Other forms of academic dishonesty are equally unacceptable. See additional information in SCampus and university policies on scientific misconduct, policy.usc.edu/scientific-misconduct.

Support Systems:

Counseling and Mental Health - (213) 740-9355 – 24/7 on call
studenthealth.usc.edu/counseling
Free and confidential mental health treatment for students, including short-term psychotherapy, group counseling, stress fitness workshops, and crisis intervention.

National Suicide Prevention Lifeline - 1 (800) 273-8255 – 24/7 on call
suicidepreventionlifeline.org
Free and confidential emotional support to people in suicidal crisis or emotional distress 24 hours a day, 7 days a week.

Relationship and Sexual Violence Prevention Services (RSVP) - (213) 740-9355(WELL), press "0" after hours – 24/7 on call
studenthealth.usc.edu/sexual-assault
Free and confidential therapy services, workshops, and training for situations related to gender-based harm.

Office of Equity and Diversity (OED) - (213) 740-5086 | Title IX – (213) 821-8298
equity.usc.edu, titleix.usc.edu
Information about how to get help or help someone affected by harassment or discrimination, rights of protected classes, reporting options, and additional resources for students, faculty, staff, visitors, and applicants.

Reporting Incidents of Bias or Harassment - (213) 740-5086 or (213) 821-8298
usc-advocate.symplicity.com/care_report
Avenue to report incidents of bias, hate crimes, and microaggressions to the Office of Equity and Diversity |Title IX for appropriate investigation, supportive measures, and response.

The Office of Disability Services and Programs - (213) 740-0776
dsp.usc.edu
Support and accommodations for students with disabilities. Services include assistance in providing readers/notetakers/interpreters, special accommodations for test taking needs, assistance with architectural barriers, assistive technology, and support for individual needs.

USC Campus Support and Intervention - (213) 821-4710
campussupport.usc.edu
Assists students and families in resolving complex personal, financial, and academic issues adversely affecting their success as a student.

Diversity at USC - (213) 740-2101
diversity.usc.edu

Information on events, programs and training, the Provost's Diversity and Inclusion Council, Diversity Liaisons for each academic school, chronology, participation, and various resources for students.

USC Emergency - UPC: (213) 740-4321, HSC: (323) 442-1000 – 24/7 on call
dps.usc.edu, emergency.usc.edu
Emergency assistance and avenue to report a crime. Latest updates regarding safety, including ways in which instruction will be continued if an officially declared emergency makes travel to campus infeasible.

USC Department of Public Safety - UPC: (213) 740-6000, HSC: (323) 442-120 – 24/7 on call
dps.usc.edu
Non-emergency assistance or information.