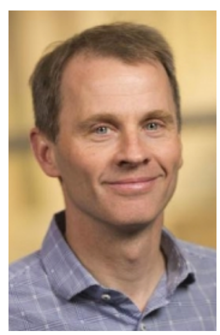
Machine Learning
Computer Sciences 760

Spring 2019

Instructors

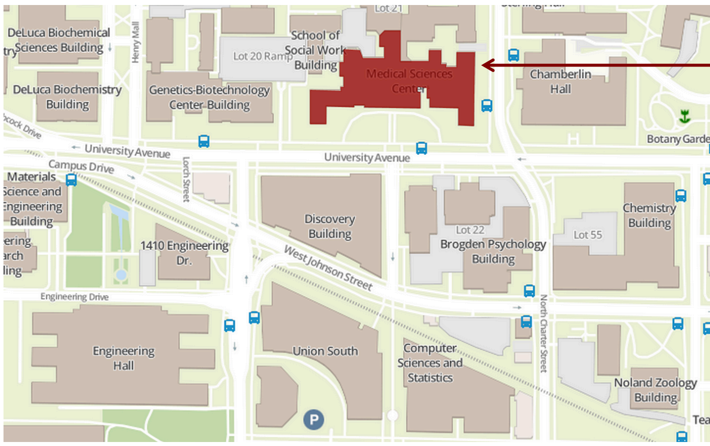


Mark Craven
- email: craven@biostat.wisc.edu
- office hours: 3-4:30 Wednesday, or by appointment
- office: 4775A Medical Sciences Center

David Page
- email: page@biostat.wisc.edu
- office hours: 1:30-3:00 Tuesday, or by appointment
- office hours room: 1157/8 Discovery Building

Finding Mark's office

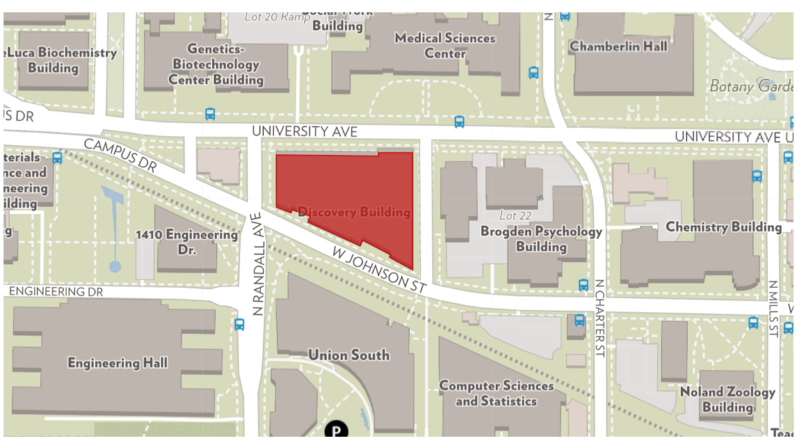
- 4775A Medical Sciences Center
- easiest to enter from Charter St. and take elevator immediately to your right



enter here

Finding David's office

- 1153/4 Discovery Building



TAs

Lichengxi “Jerry” Huang

email: lichengxi.huang@wisc.edu
office hours: 9:00-11:00 Tuesday
office: 4750 Medical Sciences Center

David Merrell

email: dmerrell@cs.wisc.edu
office hours: 3:00-5:00 Monday
office: 4750 Medical Sciences Center

The course web site

we’ll plan to use Canvas for nearly everything
https://canvas.wisc.edu/courses/127602

- syllabus
- grades
- lecture notes
- homework assignments
- homework submissions

we’ll use Piazza for questions (see the link at Canvas)

Monday, Wednesday *and* Friday?

- we'll have 26 lectures in all, just like a standard TR class
- most weeks we won't meet on Fridays
- but we will meet for the first four Fridays
- we may meet on a few other Fridays, but we'll let you know well in advance
- ***see the schedule on Canvas***

Expected background

- CS 540 (Intro to Artificial Intelligence) or equivalent
  - search
  - first-order logic
  - unification
  - deduction
- good programming skills
- basics of probability
- calculus, including partial derivatives

Course requirements

- daily in-class exercises: ~15%
- 5 homework assignments: ~35%
  - programming
  - computational experiments (e.g. measure the effect of varying parameter x in algorithm y)
  - some written exercises
- final exam: ~30%
- group project (4-5 students per group): ~20%

TopHat for in-class exercises

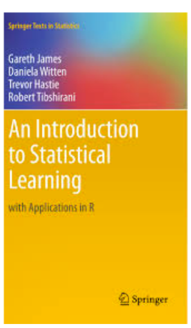
- we will use TopHat for daily in-class quizzes/exercises
- grading will be based on participation
- each student will have to set up an account and purchase a subscription ($16 for the semester, I think)
- more information at the Canvas site

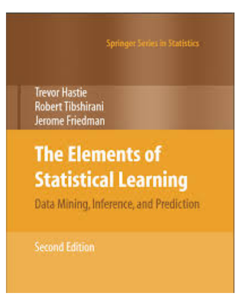
Programming assignments

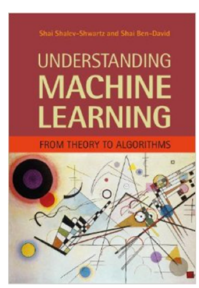
- you will be required to use Python for the programming assignments
- the advantage of this restriction is that
  - we will provide code for reading data, printing output
  - your effort will be focused on the interesting issues
- programs must be callable from the command line
- programs must run on the CS department Linux servers

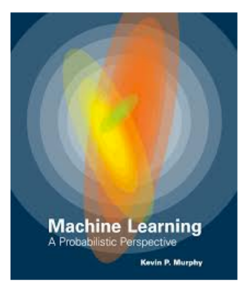
Course readings

- all readings will be on-line, no need to buy a book
- some will be from ML books









- additional on-line articles, surveys, and chapters

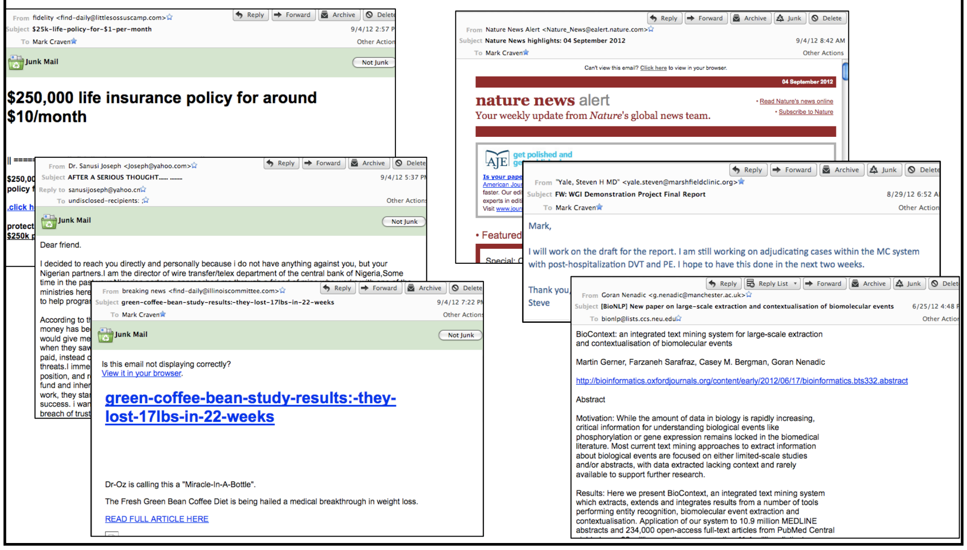
Learning objectives

1. Students will understand what a learning system should do
2. Students will distinguish among a variety of learning settings: supervised learning, unsupervised learning, reinforcement learning, active learning
3. Students will employ a broad toolbox of machine-learning methods: decision trees, nearest neighbor, linear and logistic regression, neural nets, Bayesian networks, SVMs, ensemble methods
4. Students will understand fundamental underlying theory: bias-variance tradeoff, PAC learning, mistake-bound theory
5. Students will know how to characterize how well learning systems work, and they will employ sound experimental methodology for evaluating learning systems: cross validation, ROC and PR curves, hypothesis testing

What is machine learning?

- the study of algorithms that
  improve their performance P
  at some task T
  with experience E
- to have a well defined learning task, we must specify: $< P, T, E >$

ML example: spam filtering

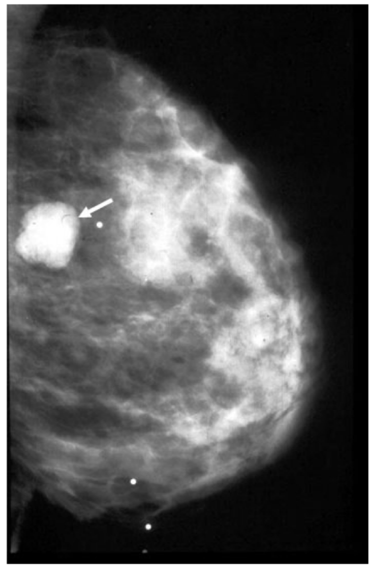


ML example: spam filtering

- T : given new mail message, classify as spam vs. other
- P : minimize misclassification costs
- E : previously classified (filed) messages

ML example: mammography

[Burnside et al., *Radiology* 2009]



ML example: mammography

- T : given new mammogram, classify each abnormality as benign vs. malignant
- P : minimize misclassification costs
- E : previously encountered patient histories (mammograms + subsequent outcomes)

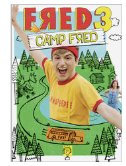
ML example: predictive text input

ML example: predictive text input

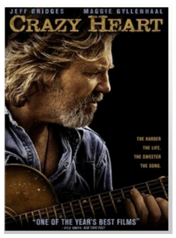
- T : given (partially) typed word, predict the word the user intended to type
- P : minimize misclassifications
- E : words previously typed by the user
(+ lexicon of common words + knowledge of keyboard layout)

domain knowledge

ML example: Netflix Prize



Our best guess for Mark:



Our best guess for Mark:

ML example: Netflix

- T : given a user/movie pair, predict the user's rating (1-5 stars) of the movie
- P : minimize difference between predicted and actual rating
- E : histories of previously rated movies (user/movie/rating triples)

ML example: reinforcement learning to control an autonomous helicopter

video of Stanford University autonomous helicopter from http://heli.stanford.edu/

ML example: autonomous helicopter

- T : given a measurement of the helicopter's current state (orientation sensor, GPS, cameras), select an adjustment of the controls
- P : maximize reward (intended trajectory + penalty function)
- E : state, action and reward triples from previous demonstration flights

Assignment

- check out detailed syllabus on Canvas
- do the readings listed for today’s, Monday’s lectures
- set up TopHat account
- set up Piazza account