Overview

Learning from data

- **Machine learning**: study of computational mechanisms that "learn" from data in order to make predictions and decisions.

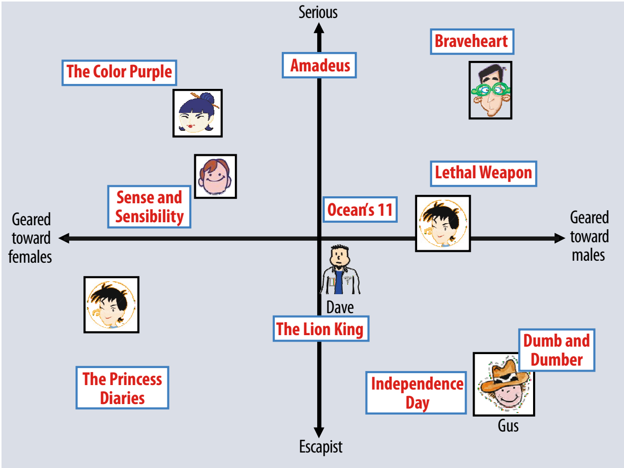
Example 1: image classification

- Birdwatcher takes pictures of birds, organizes photos by species.
- **Goal**: automatically recognize bird species in new photos.

Indigo bunting

Example 2: recommender system

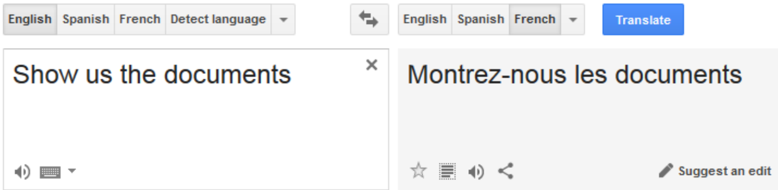
- Netflix users watch movies and provide ratings.
- **Goal**: predict the rating a user will provide on a movie not yet watched.
- (**Real goal**: keep users paying customers.)



(Graphic is from Koren, Bell, and Volinsky.)

Example 3: machine translation

- Linguists provide translations of all English language books into French, sentence-by-sentence.
- **Goal**: automatically translate any English sentence into French.



Example 4: personalized medicine

- Physician attends to patients, prescribes treatments, and observes health outcomes (e.g., recovery, death).
- **Goal**: prescribe personalized treatment for patient that delivers best possible health outcomes.

Basic setting

Data: labeled examples

$$(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \ldots, (\boldsymbol{x}_n, y_n) \ \in \ \text{Inputs} \times \text{Labels}\, \mathcal{X} \times \mathcal{Y}$$

where

- each input $\boldsymbol{x}_i$ is a description of an instance (e.g., image, (user,movie), sentence, patient), and
- each corresponding label y_i is an annotation relevant to the task (typically not easy to automatically obtain).

Goal: "learn" a function

$$\hat{f} \colon \text{Inputs} \to \text{Actions}\, \hat{f} \colon \mathcal{X} \to \mathcal{A}$$

from the data, such that for a new input $\boldsymbol{x}$ (usually without seeing its corresponding label y), the action $\hat{f}(\boldsymbol{x})$ is a "good" action.

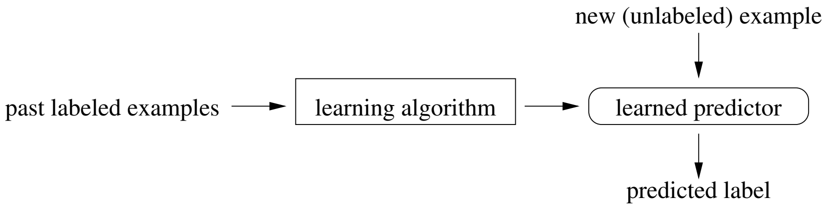
Typically, for a prediction problem, we have $\text{Actions} = \text{Labels}\ \mathcal{A} = \mathcal{Y}$ (i.e., we want the function to *predict* the labels of new inputs).

Prediction problems

- **Goal**: "learn" a prediction function (*predictor*)

$$\hat{f} \colon \text{Inputs} \to \text{Labels}$$

that provides the labels of new inputs (i.e., new *unlabeled examples*).



Why might this be possible?

Basic issues

1. What information should be recorded in the inputs, and how should they be represented?
2. What kinds of prediction functions should consider?
3. How should data be used to select a predictor?
4. How can we evaluate whether "learning" was successful?

Special case: binary classification


1


0

$\mathcal{Y} = \{0, 1\}$ (e.g., is it an indigo bunting or not)

Why is this hard?

1. Only have labels for $\{\boldsymbol{x}_i\}_{i=1}^n$, which together comprpise a miniscule fraction of the input space $\mathcal{X}$.
2. Relationship between input $\boldsymbol{x}$ and correct label $y \in \mathcal{Y}$ may be complicated, possibly ambiguous/non-deterministic!
3. Can be many functions that perfectly match inputs to labels on $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$. Which should we pick?

Machine learning in context

Intelligent systems

- **Goal**: robust system with "intelligent" / "human-like" behavior
  - **Often**: hard-coded solution too complex, not robust, sub-optimal
- How do we learn from past experiences to perform well in the future?

Algorithmic statistics

- **Goal**: statistical analysis of large, complex data sets
  - **Past**: ≤ 100 data points of two variables.
    Data collection and statistical analysis done by hand/eye.
  - **Now**: several million data and variables, collected by high-throughput automatic processes.
- How can we automate statistical analysis for modern applications?

Business application example

(Example adapted from `nlpers.blogspot.com/2016/08/debugging-machine-learning.html`)

Extracting the machine learning problem

- **Goal**: increase revenue
- **Sub-goal**: improve click-through rate on online ads
- **Sub-sub-goal**: improve prediction of click-through rate for ads based on user/website context

Approach:

1. **collect data** by logging user-ad interactions on website
2. determine **representation** for the interactions
3. decide on **learning algorithm**
4. **apply and evaluate** learning algorithm on data
5. **test** in live system

Topics for this course

Main topics:

1. Non-parametric methods (e.g., nearest neighbors, decision trees)
2. Parametric methods (e.g., generative models, linear & non-linear models)
3. Reductions (e.g., boosting, multi-class $\Rightarrow$ binary)
4. Regression (e.g., least squares, Lasso)
5. Representation learning (e.g., mixture models, collaborative filtering)

Major themes:

1. Principles of *supervised machine learning* (for prediction problems)
2. Algorithmic techniques for machine learning (statistical modeling, optimization, and reductions)
3. Some well-weathered machine learning algorithms and models

Sample of other topics in machine learning

Advanced issues

- Distributed learning
- Causal inference
- Privacy and fairness

Other models of learning

- Semi-supervised learning
- Online learning
- Reinforcement learning

Application areas

- Natural language processing
- Computer vision
- Computational advertising

Modes of study

- Mathematical analysis
- Cross-domain evaluations
- End-to-end application study

Prerequisites

Mathematical prerequisites

- Linear algebra (e.g., vector spaces, orthogonality, spectral decomposition)
- Probability (e.g., conditional probability, independence, random variables)
- Multivariate calculus (e.g., limits, Taylor expansion, gradients)
- Basic algorithms and data structures (e.g., correctness and efficiency analysis, dynamic programming)

Computational prerequisites

- Regular access to and ability to program in Python or MATLAB.

  MATLAB is available for download for SEAS students:
  `http://portal.seas.columbia.edu/matlab/`

  Coupons for Google Cloud infrastructure available. (Very easy to use!)

Course requirements

1. Complete assigned reading (posted on website) before each lecture.
2. Attend lecture (either in-person or via CVN).
   Lecture slides posted on course website shortly after each lecture.
3. Complete ∼seven homework assignments (theory & programming): 40%.
4. Complete two in-class exams (Oct 24, Dec 12): 30% each.

Resources

http://www.cs.columbia.edu/~djhsu/coms4771-f16/

Course staff

- **Instructor**: Prof. Daniel Hsu
- **Teaching assistants**: Eugene, Patanjali, and Siddharth (see course website)
- **Office hours, course e-mail, online forum (Piazza)**: see course website

Class policies

- No late assignments accepted without valid medical/family emergency, as authenticated by your academic adviser (and a physician, if applicable).
- No make-up exams.
  In case of a valid medical/family emergency (authenticated as above), your grade composition will be adjusted.
- Add/drop deadlines: your own responsibility.
  http://registrar.columbia.edu/content/post-change-program-adddrop-period
  Note: if you're going to drop, please do it now.
- Disability services: make arrangements for accommodations and other services within first two weeks of class.
  https://health.columbia.edu/disability-services

Academic rules of conduct

- See course website, and also Academic Honesty policy of the Computer Science Department.
  http://www.cs.columbia.edu/education/honesty
- It is your responsibility to understand the distinction between cheating and allowed cooperation/collaboration.
  If ever in doubt, ask the instructor.
- Any violation will result in a penalty to be assessed at the instructor's discretion.
  This may include receiving a zero grade for the assignment in question AND a failing grade for the whole course, even for the first infraction.

Homework 0

First homework assignment ("Homework 0") due Monday.

- **Required**; submit on Courseworks.
- Partly intended to help you "page-in" mathematical prerequisites.
- If you have difficulty with the assignment, **it is likely that much of the course will be especially difficult.**
- **If you cannot complete the assignment, you are strongly advised to drop the course.**

Key takeaways

1. Examples of machine learning problems and why they are challenging.
2. Setup of simple prediction and classification problems.
3. Course information.