6.867 Machine learning: lecture 1

Tommi S. Jaakkola
MIT CSAIL
tommi@csail.mit.edu

6.867 Machine learning: administrivia

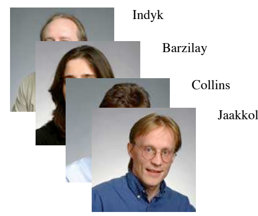
- Course staff (6867-staff@lists.csail.mit.edu)
  - Prof. Tommi Jaakkola (tommi@csail.mit.edu)
  - Adrian Corduneanu (adrianc@mit.edu)
  - Biswajit (Biz) Bose (cielbleu@mit.edu)
- General info
  - lectures MW 2.30-4pm in 32-141
  - tutorials/recitations, initially F11-12.30 (4-145) / F2.30-4 (4-159)
  - website http://www.ai.mit.edu/courses/6.867/
- Grading
  - midterm (15%), final (25%)
  - 5 (≈ bi-weekly) problem sets (30%)
  - final project (30%)

Why learning?

- Example problem: face recognition

Why learning?

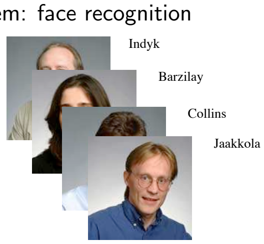
- Example problem: face recognition



Training data: a collection of images and labels (names)

Why learning?

- Example problem: face recognition

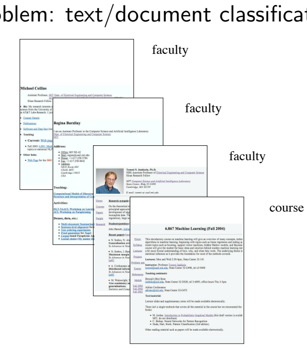


Training data: a collection of images and labels (names)

Evaluation criterion: correct labeling of new images

Why learning?

- Example problem: text/document classification

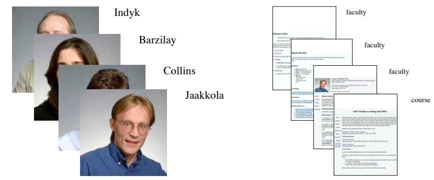


  - a few labeled training documents (webpages)
  - goal to label yet unseen documents

Why learning?

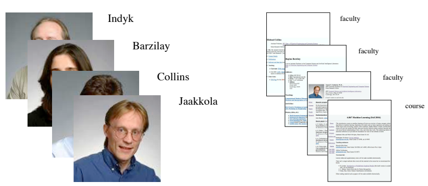
- There are already a number of applications of this type
  - face, speech, handwritten character recognition
  - fraud detection (e.g., credit card)
  - recommender problems (e.g., which movies/products/etc you'd like)
  - annotation of biological sequences, molecules, or assays
  - market prediction (e.g., stock/house prices)
  - finding errors in computer programs, computer security
  - defense applications
  - etc

Learning



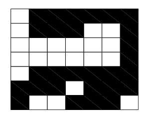
- Steps
  - entertain a (biased) set of possibilities (hypothesis class)
  - adjust predictions based on available examples (estimation)
  - rethink the set of possibilities (model selection)

Learning



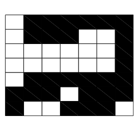
- Steps
  - entertain a (biased) set of possibilities (hypothesis class)
  - adjust predictions based on available examples (estimation)
  - rethink the set of possibilities (model selection)
- Principles of learning are "universal"
  - society (e.g., scientific community)
  - animal (e.g., human)
  - machine

Learning, biases, representation

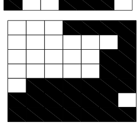


"yes"

Learning, biases, representation

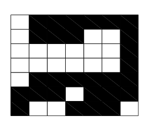


"yes"

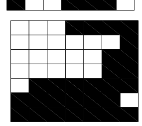


"yes"

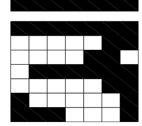
Learning, biases, representation



"yes"

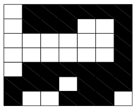


"yes"



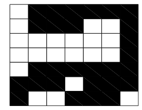
"no"

Learning, biases, representation

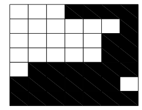


"yes"

Learning, biases, representation

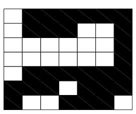


"yes"

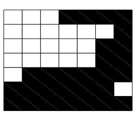


"yes"

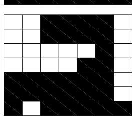
Learning, biases, representation



"yes"



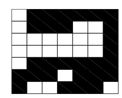
"yes"



"no" (oops)

Representation

- There are many ways of presenting the same information



0111111001110010000000010000000100111111011101111110011101111110001

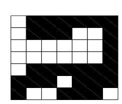
- The choice of representation may determine whether the learning task is very easy or very difficult

Representation

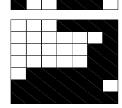
0111111001110010000000010000000100111111011101111110011101111110001 "yes"
0001111110000001100000111000001100111111011111100111111111011111111 "yes"
1111111000000110000011000111111100000011110000011111100011011100 01 "no"

Representation

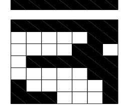
0111111001110010000000010000000100111111011101111110011101111110001 "yes"
0001111110000001100000111000001100111111011111100111111111011111111 "yes"
1111111000000110000011000111111100000011110000011111100011011100 01 "no"



"yes"



"yes"

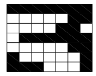


"no"

Hypothesis class

- Representation: examples are binary vectors of length $d = 64$

$$\mathbf{x} = [111\ldots0001]^T \ =$$



and labels $y \in \{-1, 1\}$ ("no","yes")

- The mapping from examples to labels is a "linear classifier"

$$\hat{y} = \text{sign}\,(\,\theta\cdot\mathbf{x}\,) = \text{sign}\,(\,\theta_1 x_1 + \ldots + \theta_d x_d\,)$$

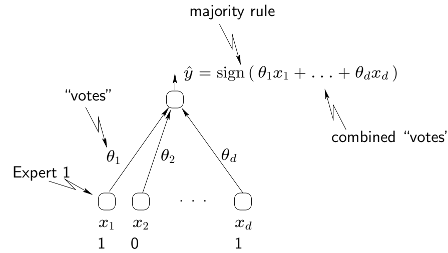
where θ is a vector of *parameters* we have to learn from examples.

Linear classifier/experts

- We can understand the simple linear classifier

$$\hat{y} = \text{sign}\,(\,\theta\cdot\mathbf{x}\,) = \text{sign}\,(\,\theta_1 x_1 + \ldots + \theta_d x_d\,)$$

as a way of combining expert opinion (in this case simple binary features)



Estimation

$\mathbf{x}$	y
0111111001110010000000010000000100111111011101111110011101111110001	+1
0001111110000001100000111000001100111111011111100111111110000011	+1
1111111000000110000011000111111000000111100000111110001101111111	−1
. . .	. . .

- How do we adjust the parameters θ based on the labeled examples?

$$\hat{y} = \text{sign}\,(\,\theta\cdot\mathbf{x}\,)$$

Estimation

$\mathbf{x}$	y
0111111001110010000000010000000100111111011101111110011101111110001	+1
0001111110000001100000111000001100111111011111100111111110000011	+1
1111111000000110000011000111111000000111100000111110001101111111	−1
. . .	. . .

- How do we adjust the parameters θ based on the labeled examples?

$$\hat{y} = \text{sign}\,(\,\theta\cdot\mathbf{x}\,)$$

For example, we can simply refine/update the parameters whenever we make a mistake:

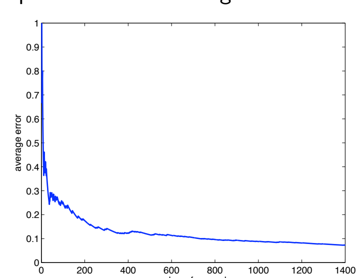
$$\theta_i \leftarrow \theta_i + y\,x_i,\ i = 1,\ldots,d \quad \text{if prediction was wrong}$$

Evaluation

- Does the simple mistake driven algorithm work?

Evaluation

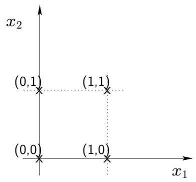
- Does the simple mistake driven algorithm work?



(average classification error as a function of the number of examples and labels seen so far)

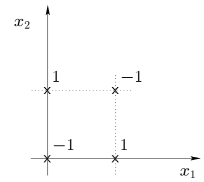
Model selection

- The simple linear classifier cannot solve all the problems (e.g., XOR)



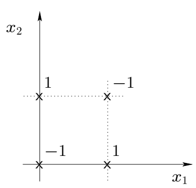
Model selection

- The simple linear classifier cannot solve all the problems (e.g., XOR)



Model selection

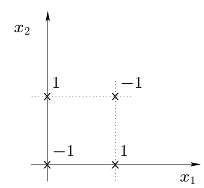
- The simple linear classifier cannot solve all the problems (e.g., XOR)



- Can we rethink the approach to do even better?

Model selection

- The simple linear classifier cannot solve all the problems (e.g., XOR)

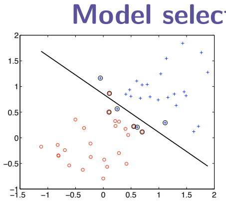


- Can we rethink the approach to do even better?

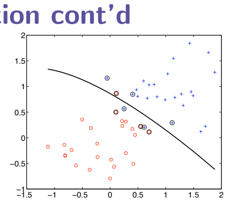
  We can, for example, add "polynomial experts"

$$\hat{y} = \text{sign}\left(\, \theta_1 x_1 + \ldots + \theta_d x_d + \theta_{12} x_1 x_2 + \ldots \right)$$

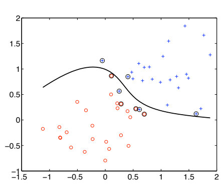
Model selection cont'd



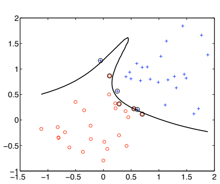
linear



2^{nd} order polynomial



4^{th} order polynomial



8^{th} order polynomial

Types of learning problems (not exhaustive)

- *Supervised* learning: explicit feedback in the form of examples and target labels
  - goal to make predictions based on examples (classify them, predict prices, etc)
- *Unsupervised* learning: only examples, no explicit feedback
  - goal to reveal structure in the observed data
- *Semi-supervised* learning: limited explicit feedback, mostly only examples
  - tries to improve predictions based on examples by making use of the additional "unlabeled" examples
- *Reinforcement* learning: delayed and partial feedback, no explicit guidance
  - goal to minimize the cost of a sequence of actions (policy)