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: face recognition

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

- 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

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"

"yes"

"no" (oops)

Representation

- There are many ways of presenting the same information

```
01111100110100000100000001001110111110001111110111111011111101111110001
```

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

```
001111001100000101111100010111111100011111101111110111111011111101111110001
```

"yes""yes"

"no"

"yes""yes"

"no"
Hypothesis class
- Representation: examples are binary vectors of length $d = 64$

\[ x = [111 \ldots 0001]^T = \]

and labels $y \in \{-1, 1\}$ ("no","yes")
- The mapping from examples to labels is a "linear classifier"

\[ \hat{y} = \text{sign} (\theta \cdot x) = \text{sign}(\theta_1 x_1 + \ldots + \theta_d x_d) \]

where $\theta$ is a vector of parameters we have to learn from examples.

Linear classifier/experts
- We can understand the simple linear classifier

\[ \hat{y} = \text{sign}(\theta \cdot x) = \text{sign}(\theta_1 x_1 + \ldots + \theta_d x_d) \]

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

Estimation
- How do we adjust the parameters $\theta$ based on the labeled examples?

\[ \hat{y} = \text{sign}(\theta \cdot x) \]

- How do we adjust the parameters $\theta$ based on the labeled examples?

\[ \hat{y} = \text{sign}(\theta \cdot x) \]

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

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

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)

![Graph showing XOR problem]

- Can we rethink the approach to do even better?

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

Model selection cont’d

- 4th order polynomial
- 5th 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)