Announcements

Assignments

- HW5
  - Due Mon, 10/26, 11:59 pm
  - Start early

Recitation

- No recitation this Friday

Educational Research

- See section added to the end of the website

Plan

Last Time

- Neural Networks
  - Calculus
  - Universal Approximation Theorem
  - Convolutional neural networks

Today

- Wrap up convolutional neural networks
- Learning Theory
  - Bias and variance
  - Learning theory model
  - Introduce PAC learning

Wrap Up Neural Networks

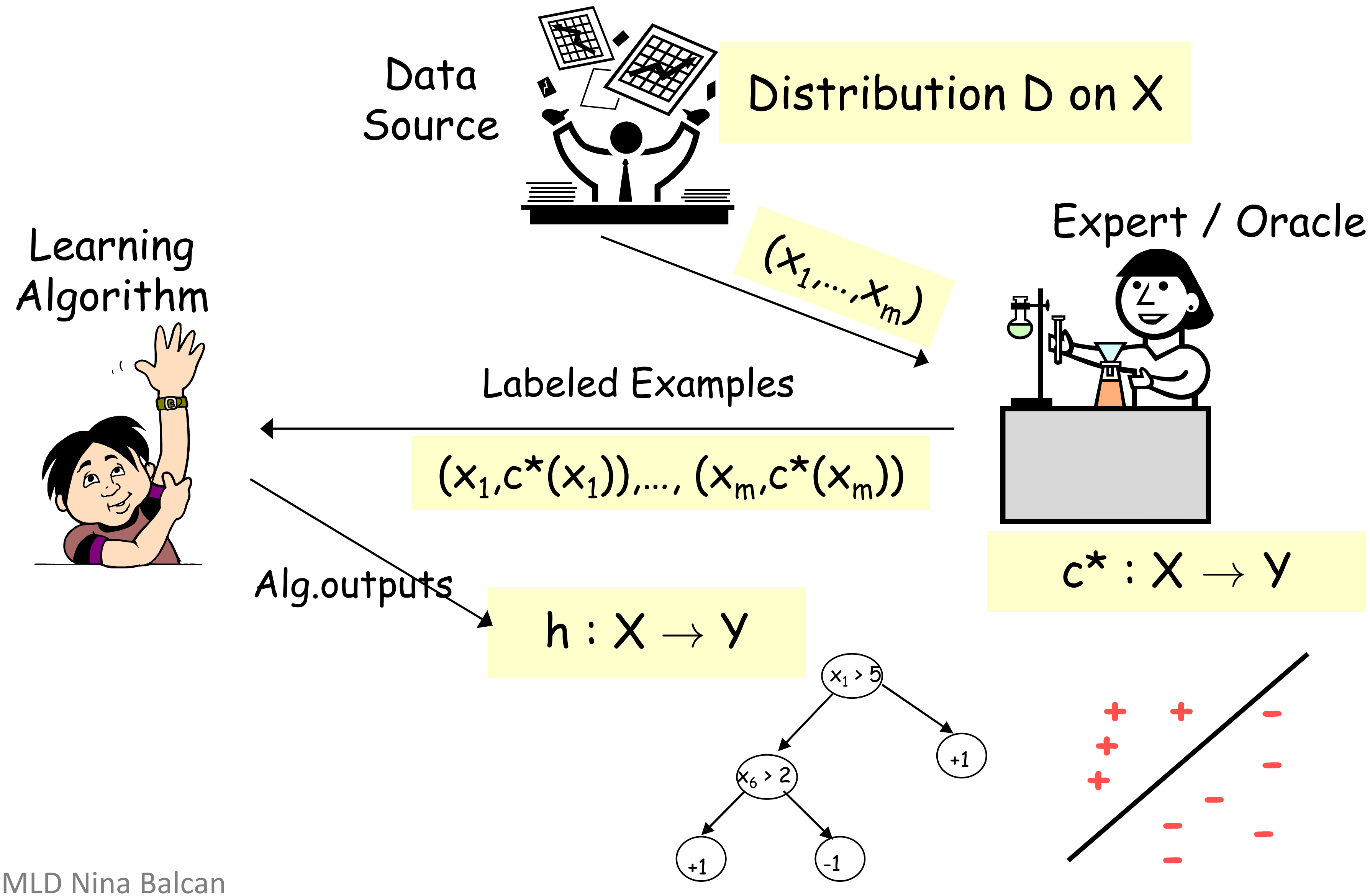
Neural network slides

Introduction to Machine Learning

Learning Theory

Instructor: Pat Virtue

Model for Supervised Learning



Slide credit: CMU MLD Nina Balcan

Learning from Training Data

We want to learn from training data

But, we also want our hypothesis function to generalize well

- How do we characterize and quantify these properties?
- Bias and variance

Bias and Variance Examples

Slide credit: Andrew Ng, Stanford

Bias and Variance Examples

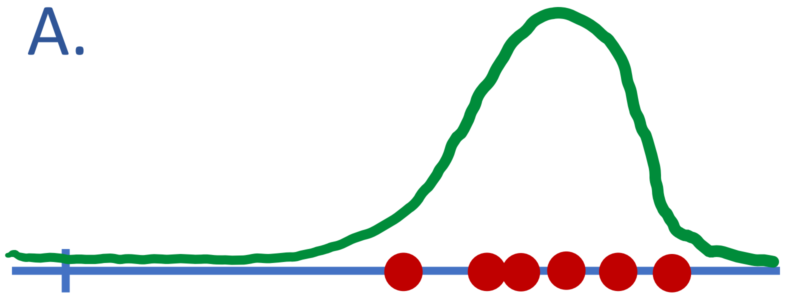
Slide credit: Andrew Ng, Stanford

Piazza Polls 1 & 2

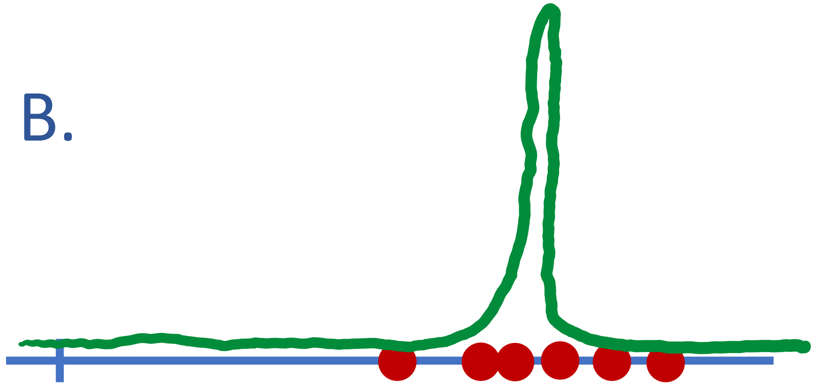
Poll 1: [SELECT TWO] Which have high variance?

Poll 2: [SELECT TWO] Which have high bias?

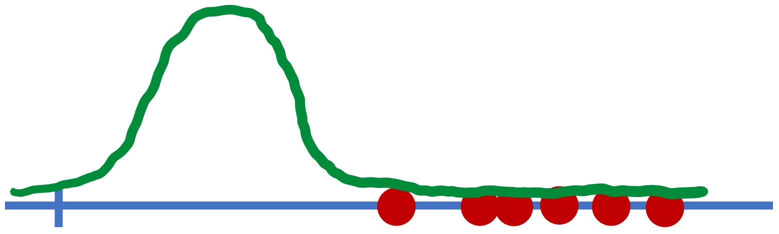
A.



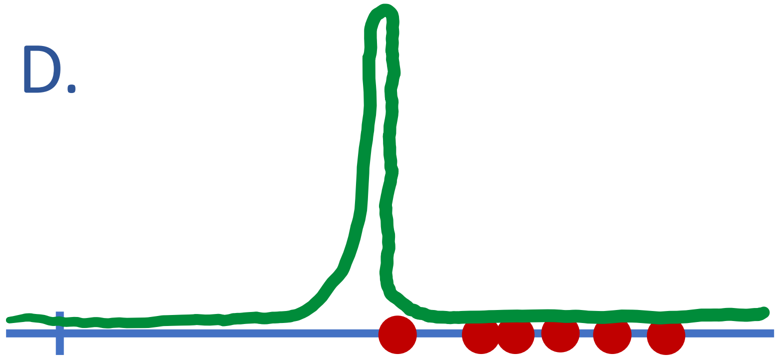
B.



C.



D.



Questions

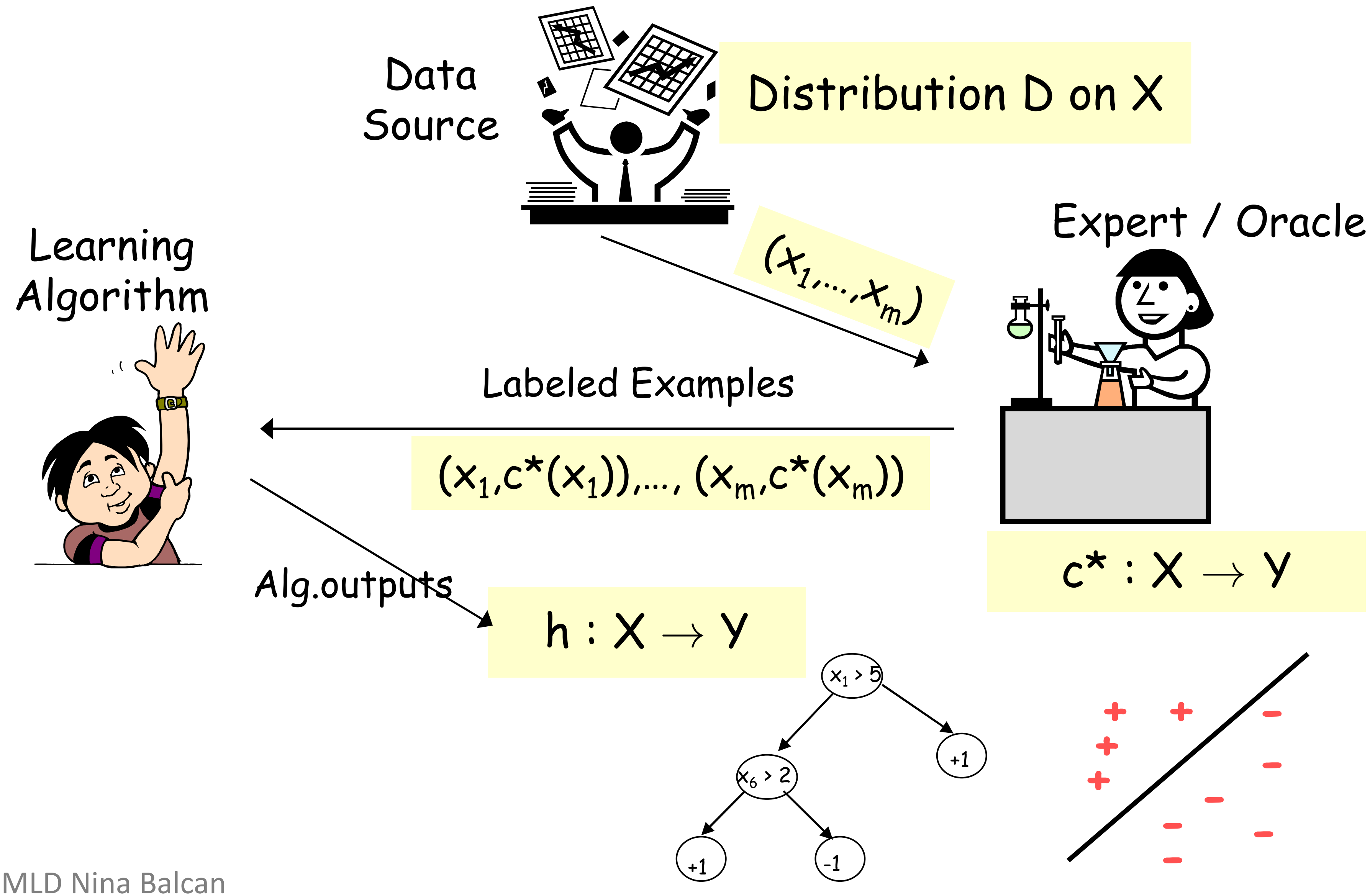
1. Given a classifier with **zero training error**, what can we say about **true error** (aka. generalization error)?
(Sample Complexity, Realizable Case)

2. Given a classifier with **low training error**, what can we say about **true error** (aka. generalization error)?
(Sample Complexity, Agnostic Case)

3. Is there a **theoretical justification for regularization** to avoid overfitting?
(Structural Risk Minimization)

Slide credit: CMU MLD Matt Gormley

Model for Supervised Learning



Slide credit: CMU MLD Nina Balcan

Two Types of Error

1. True Error (aka. **expected risk**)

$$R(h) = P_{\mathbf{x} \sim p^*(\mathbf{x})}(c^*(\mathbf{x}) \neq h(\mathbf{x}))$$

This quantity is always **unknown**

2. Train Error (aka. **empirical risk**)

$$\hat{R}(h) = P_{\mathbf{x} \sim \mathcal{S}}(c^*(\mathbf{x}) \neq h(\mathbf{x}))$$

$$= \frac{1}{N} \sum_{i=1}^{N} \mathbb{1}(c^*(\mathbf{x}^{(i)}) \neq h(\mathbf{x}^{(i)}))$$

$$= \frac{1}{N} \sum_{i=1}^{N} \mathbb{1}(y^{(i)} \neq h(\mathbf{x}^{(i)}))$$

We can **measure** this on the training data

where $\mathcal{S} = \{\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(N)})\}_{i=1}^{N}$ is the training data set, and $\mathbf{x} \sim \mathcal{S}$ denotes that $\mathbf{x}$ is sampled from the empirical distribution.

PAC / SLT Model

1. Generate instances from *unknown* distribution p^*

$$\mathbf{x}^{(i)} \sim p^*(\mathbf{x}), \; \forall i \tag{1}$$

2. Oracle labels each instance with *unknown* function c^*

$$y^{(i)} = c^*(\mathbf{x}^{(i)}), \; \forall i \tag{2}$$

3. Learning algorithm chooses hypothesis $h \in \mathcal{H}$ with low(est) training error, $\hat{R}(h)$

$$\hat{h} = \operatorname*{argmin}_h \hat{R}(h) \tag{3}$$

4. Goal: Choose an h with low generalization error $R(h)$

Three Hypotheses of Interest

The **true function** c^* is the one we are trying to learn and that labeled the training data:

$$y^{(i)} = c^*(\mathbf{x}^{(i)}),\ \forall i \tag{1}$$

The **expected risk minimizer** has lowest true error:

$$h^* = \underset{h \in \mathcal{H}}{\text{argmin}}\ R(h)$$

Question:
True or False:
h* and c* are always equal.

The **empirical risk minimizer** has lowest training error:

$$\hat{h} = \underset{h \in \mathcal{H}}{\text{argmin}}\ \hat{R}(h) \tag{3}$$

Slide credit: CMU MLD Matt Gormley