10-601 Introduction to Machine Learning

Machine Learning Department
School of Computer Science
Carnegie Mellon University

Midterm Review

+

PAC Learning

Matt Gormley
Lecture 15
Oct. 22, 2018

Outline

- Midterm Exam Logistics
- Sample Questions
- Classification and Regression: The Big Picture
- Q&A

MIDTERM EXAM LOGISTICS

Midterm Exam

- **Time / Location**
  - **Time:** Evening Exam
    Thu, Oct. 25 at 6:30pm – 9:00pm
  - **Room**: We will contact each student individually with **your room assignment**. The rooms are **not** based on section.
  - **Seats:** There will be **assigned seats**. Please arrive early.
  - Please watch Piazza carefully for announcements regarding room / seat assignments.
- **Logistics**
  - Format of questions:
    - Multiple choice
    - True / False (with justification)
    - Derivations
    - Short answers
    - Interpreting figures
    - Implementing algorithms on paper
  - No electronic devices
  - You are allowed to **bring** one 8½ x 11 sheet of notes (front and back)

Midterm Exam

- **How to Prepare**
  - Attend the midterm review lecture (right now!)
  - Review prior year's exam and solutions (we'll post them)
  - Review this year's homework problems
  - Consider whether you have achieved the "learning objectives" for each lecture / section

Midterm Exam

- **Advice (for during the exam)**
  - Solve the easy problems first (e.g. multiple choice before derivations)
    - if a problem seems extremely complicated you're likely missing something
  - Don't leave any answer blank!
  - If you make an assumption, write it down
  - If you look at a question and don't know the answer:
    - we probably haven't told you the answer
    - but we've told you enough to work it out
    - imagine arguing for some answer and see if you like it

Topics for Midterm

- Foundations
  - Probability, Linear Algebra, Geometry, Calculus
  - MLE
  - Optimization
- Important Concepts
  - Regularization and Overfitting
  - Experimental Design
- Classifiers
  - Decision Tree
  - KNN
  - Perceptron
  - Logistic Regression
- Regression
  - Linear Regression
- Feature Learning
  - Neural Networks
  - Basic NN Architectures
  - Backpropagation
- Learning Theory
  - PAC Learning

SAMPLE QUESTIONS

Matching Game

Goal: Match the Algorithm to its Update Rule

1. SGD for Logistic Regression

$$h_{\boldsymbol{\theta}}(\mathbf{x}) = p(y|x)$$

2. Least Mean Squares

$$h_{\boldsymbol{\theta}}(\mathbf{x}) = \boldsymbol{\theta}^T \mathbf{x}$$

3. Perceptron (next lecture)

$$h_{\boldsymbol{\theta}}(\mathbf{x}) = \text{sign}(\boldsymbol{\theta}^T \mathbf{x})$$

4.

$$\theta_k \leftarrow \theta_k + (h_{\boldsymbol{\theta}}(\mathbf{x}^{(i)}) - y^{(i)})$$

5.

$$\theta_k \leftarrow \theta_k + \frac{1}{1 + \exp \lambda(h_{\boldsymbol{\theta}}(\mathbf{x}^{(i)}) - y^{(i)})}$$

6.

$$\theta_k \leftarrow \theta_k + \lambda(h_{\boldsymbol{\theta}}(\mathbf{x}^{(i)}) - y^{(i)}) x_k^{(i)}$$

A. 1=5, 2=4, 3=6
B. 1=5, 2=6, 3=4
C. 1=6, 2=4, 3=4
D. 1=5, 2=6, 3=6
E. 1=6, 2=6, 3=6

Sample Questions

1.4 Probability

Assume we have a sample space Ω. Answer each question with **T** or **F**.

(a) [1 pts.] **T or F:** If events A, B, and C are disjoint then they are independent.

(b) [1 pts.] **T or F:** $P(A|B) \propto \frac{P(A)P(B|A)}{P(A|B)}$. (The sign '$\propto$' means 'is proportional to')

Sample Questions

4 K-NN [12 pts]

Now we will apply K-Nearest Neighbors using Euclidean distance to a binary classification task. We assign the class of the test point to be the class of the majority of the k nearest neighbors. A point can be its own neighbor.

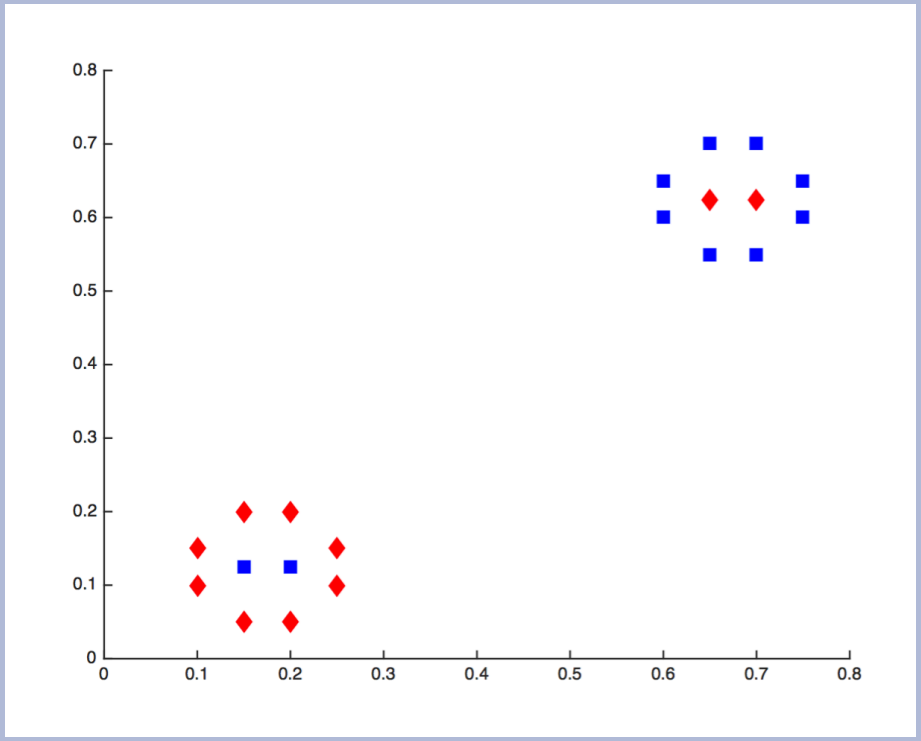


Figure 5

3. **[2 pts]** What value of k minimizes leave-one-out cross-validation error for the dataset shown in Figure 5? What is the resulting error?

Sample Questions

3.1 Linear regression

Consider the dataset S plotted in Fig. 1 along with its associated regression line. For each of the altered data sets S^{new} plotted in Fig. 3, indicate which regression line (relative to the original one) in Fig. 2 corresponds to the regression line for the new data set. Write your answers in the table below.

Dataset	(a)	(b)	(c)	(d)	(e)
Regression line					

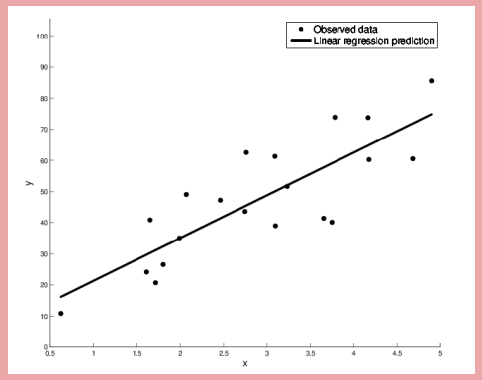
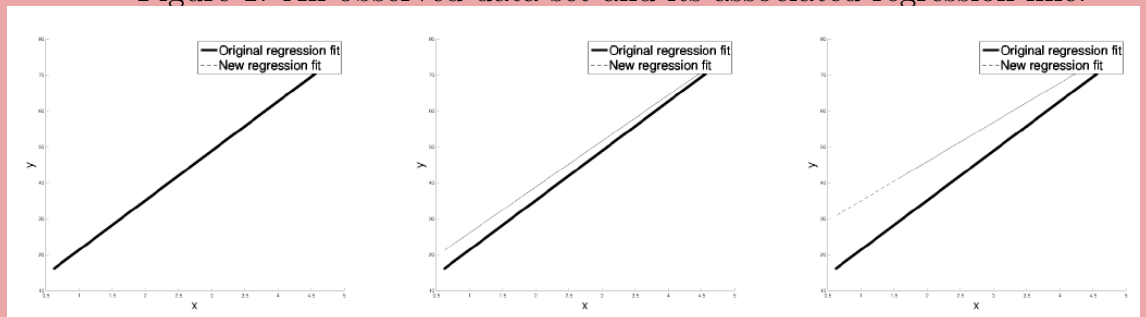


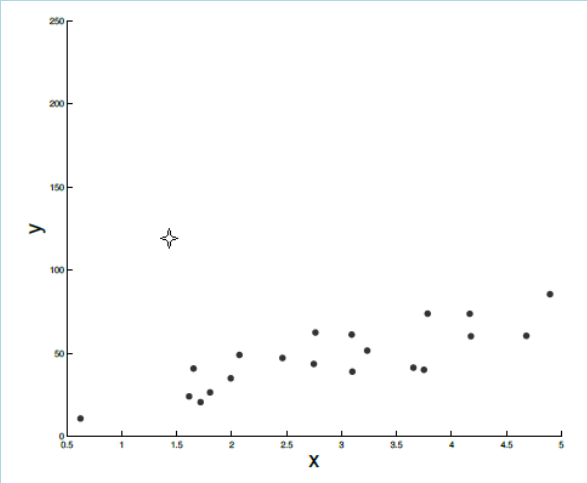
Figure 1: An observed data set and its associated regression line.



(a) Old and new regression lines. (b) Old and new regression lines. (c) Old and new regression lines.

Figure 2: New regression lines for altered data sets S^{new}.

Dataset



(a) Adding one outlier to the original data set.

Sample Questions

3.1 Linear regression

Consider the dataset S plotted in Fig. 1 along with its associated regression line. For each of the altered data sets S^{new} plotted in Fig. 3, indicate which regression line (relative to the original one) in Fig. 2 corresponds to the regression line for the new data set. Write your answers in the table below.

Dataset	(a)	(b)	(c)	(d)	(e)
Regression line					

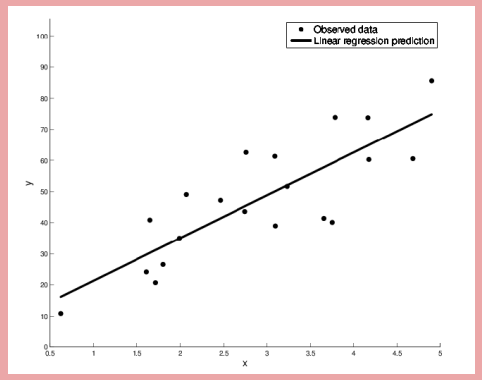
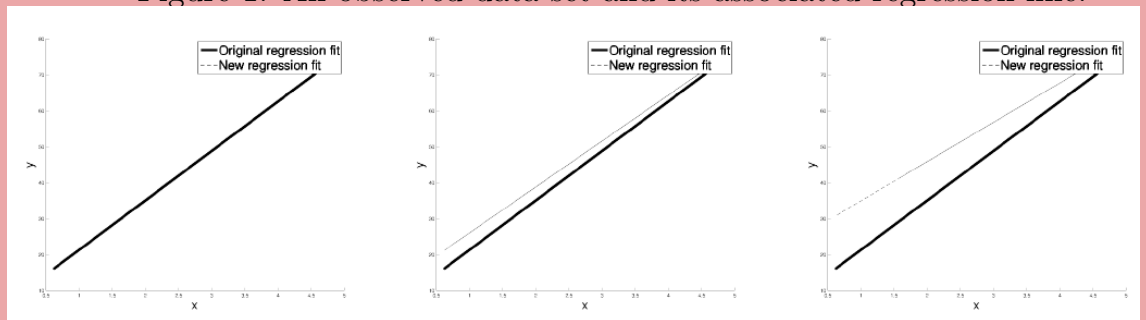


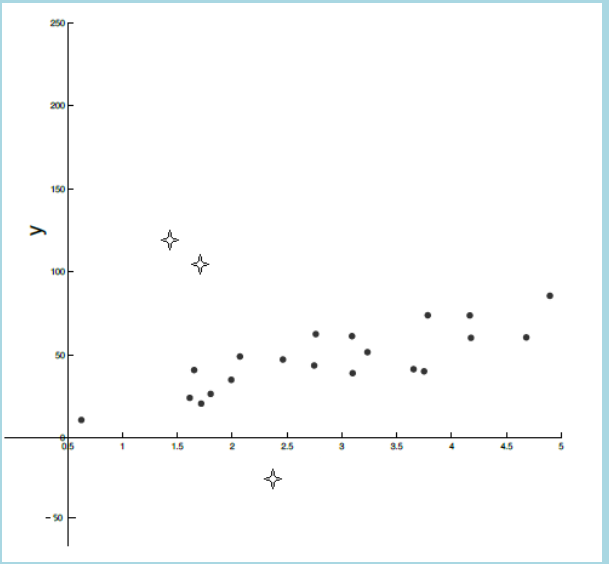
Figure 1: An observed data set and its associated regression line.



(a) Old and new regression lines. (b) Old and new regression lines. (c) Old and new regression lines.

Figure 2: New regression lines for altered data sets S^{new}.

Dataset



(c) Adding three outliers to the original data set. Two on one side and one on the other side.

Sample Questions

3.1 Linear regression

Consider the dataset S plotted in Fig. 1 along with its associated regression line. For each of the altered data sets S^{new} plotted in Fig. 3, indicate which regression line (relative to the original one) in Fig. 2 corresponds to the regression line for the new data set. Write your answers in the table below.

Dataset	(a)	(b)	(c)	(d)	(e)
Regression line					

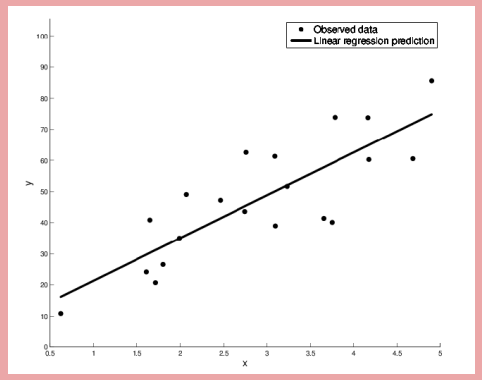
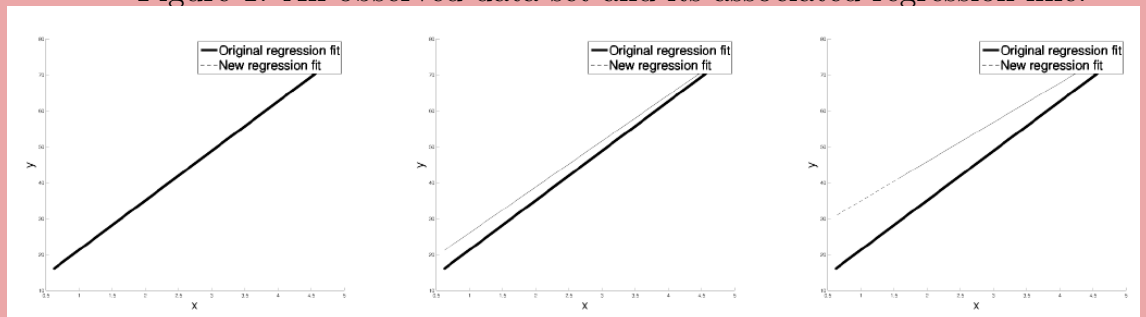


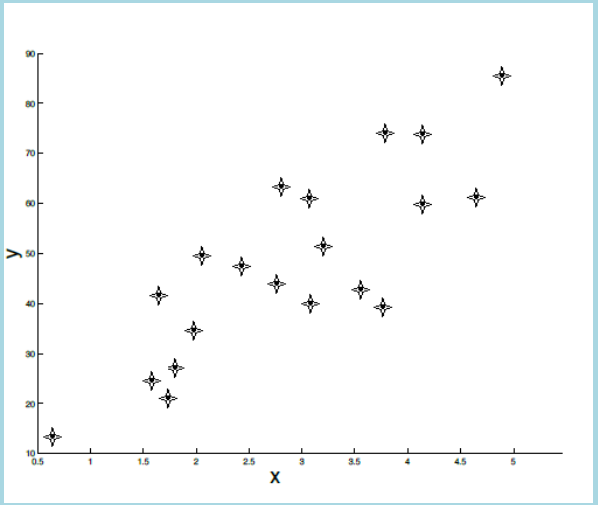
Figure 1: An observed data set and its associated regression line.



(a) Old and new regression lines. (b) Old and new regression lines. (c) Old and new regression lines.

Figure 2: New regression lines for altered data sets S^{new}.

Dataset



(d) Duplicating the original data set.

Sample Questions

3.1 Linear regression

Consider the dataset S plotted in Fig. 1 along with its associated regression line. For each of the altered data sets S^{new} plotted in Fig. 3, indicate which regression line (relative to the original one) in Fig. 2 corresponds to the regression line for the new data set. Write your answers in the table below.

Dataset	(a)	(b)	(c)	(d)	(e)
Regression line					

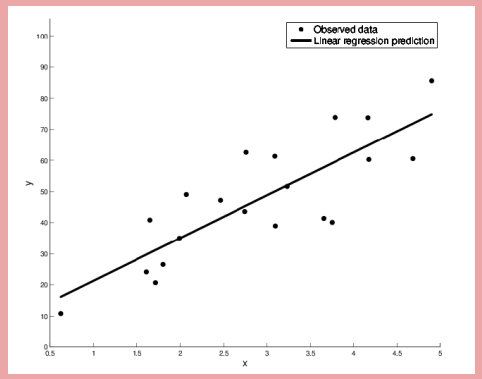
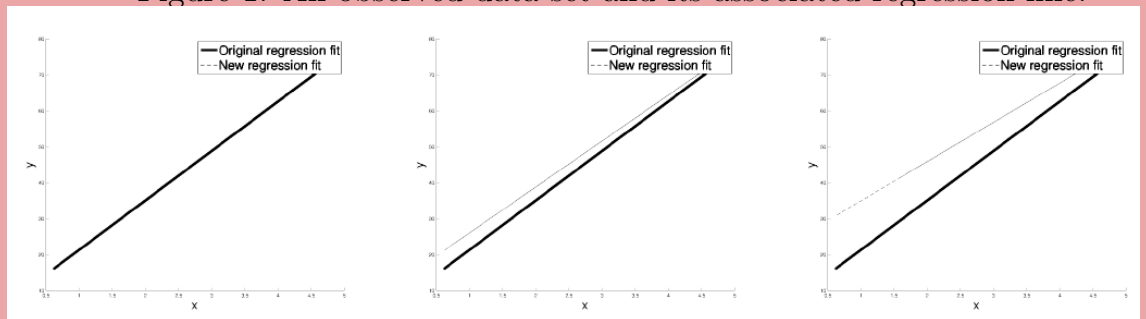


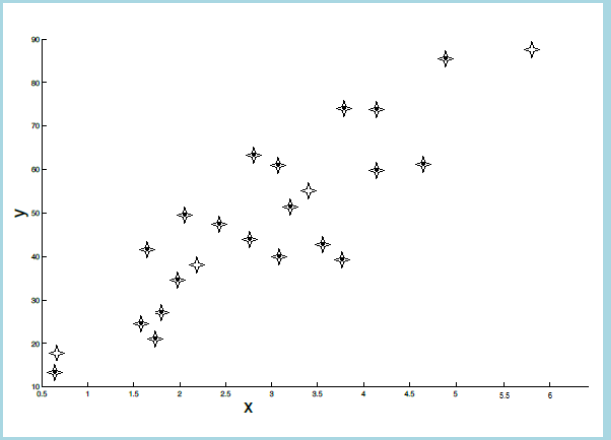
Figure 1: An observed data set and its associated regression line.



(a) Old and new regression lines. (b) Old and new regression lines. (c) Old and new regression lines.

Figure 2: New regression lines for altered data sets S^{new}.

Dataset



(e) Duplicating the original data set and adding four points that lie on the trajectory of the original regression line.

Sample Questions

3.2 Logistic regression

Given a training set $\{(x_i, y_i), i = 1, \ldots, n\}$ where $x_i \in \mathbb{R}^d$ is a feature vector and $y_i \in \{0, 1\}$ is a binary label, we want to find the parameters $\hat{w}$ that maximize the likelihood for the training set, assuming a parametric model of the form

$$p(y = 1|x; w) = \frac{1}{1 + \exp(-w^T x)}.$$

The conditional log likelihood of the training set is

$$\ell(w) = \sum_{i=1}^{n} y_i \log p(y_i, |x_i; w) + (1 - y_i) \log(1 - p(y_i, |x_i; w)),$$

and the gradient is

$$\nabla \ell(w) = \sum_{i=1}^{n} (y_i - p(y_i|x_i; w))x_i.$$

(b) [5 pts.] What is the form of the classifier output by logistic regression?

(c) [2 pts.] **Extra Credit:** Consider the case with binary features, i.e, $x \in \{0, 1\}^d \subset \mathbb{R}^d$, where feature x_1 is rare and happens to appear in the training set with only label 1. What is $\hat{w}_1$? Is the gradient ever zero for any finite w? Why is it important to include a regularization term to control the norm of $\hat{w}$?

Samples Questions

2.1 Train and test errors

In this problem, we will see how you can debug a classifier by looking at its train and test errors. Consider a classifier trained till convergence on some training data $\mathcal{D}^{\text{train}}$, and tested on a separate test set $\mathcal{D}^{\text{test}}$. You look at the test error, and find that it is very high. You then compute the training error and find that it is close to 0.

1. **[4 pts]** Which of the following is expected to help? Select all that apply.

   (a) Increase the training data size.

   (b) Decrease the training data size.

   (c) Increase model complexity (For example, if your classifier is an SVM, use a more complex kernel. Or if it is a decision tree, increase the depth).

   (d) Decrease model complexity.

   (e) Train on a combination of $\mathcal{D}^{\text{train}}$ and $\mathcal{D}^{\text{test}}$ and test on $\mathcal{D}^{\text{test}}$

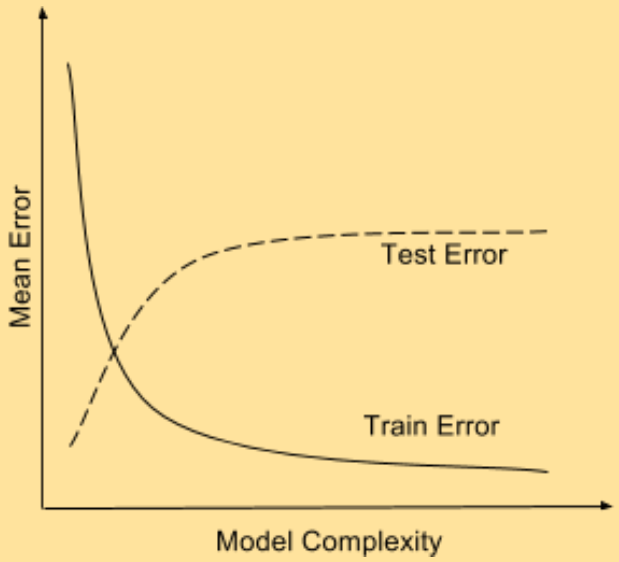
   (f) Conclude that Machine Learning does not work.

Samples Questions

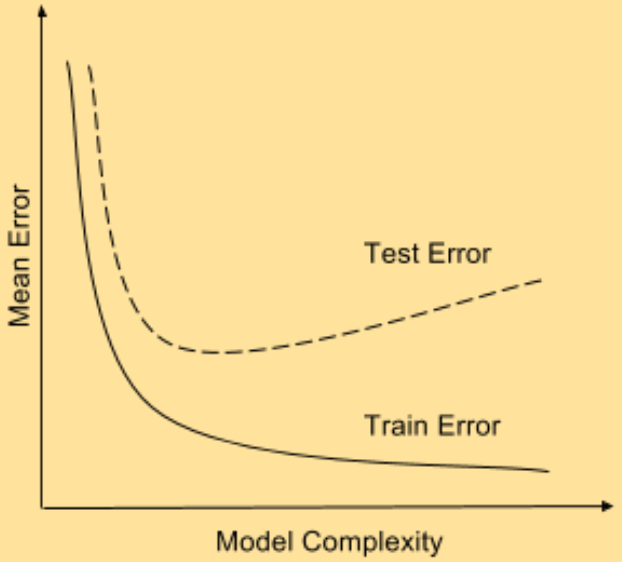
2.1 Train and test errors

In this problem, we will see how you can debug a classifier by looking at its train and test errors. Consider a classifier trained till convergence on some training data $\mathcal{D}^{\text{train}}$, and tested on a separate test set $\mathcal{D}^{\text{test}}$. You look at the test error, and find that it is very high. You then compute the training error and find that it is close to 0.

4. **[1 pts]** Say you plot the train and test errors as a function of the model complexity. Which of the following two plots is your plot expected to look like?



(a)



(b)

Sample Questions

4.1 True or False

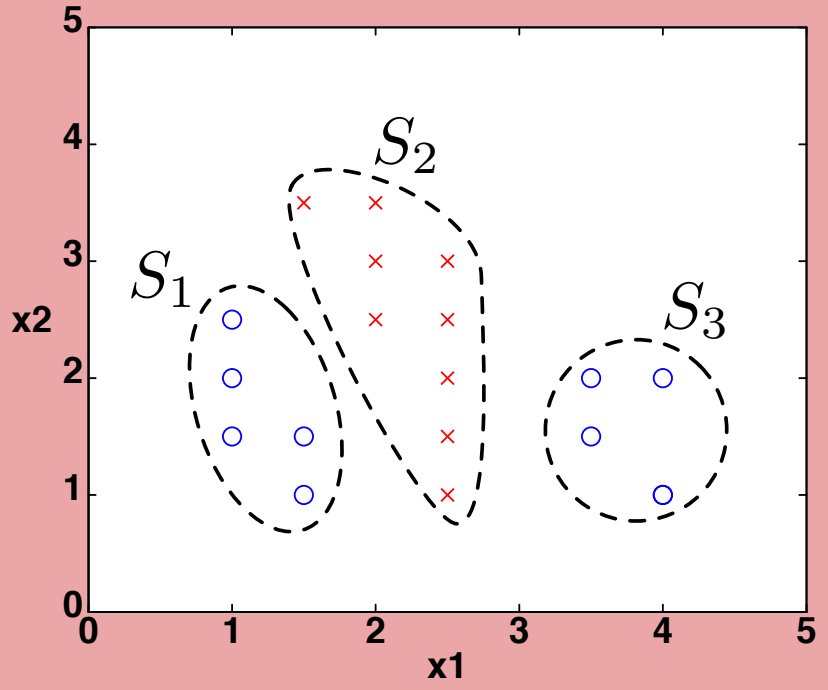
Answer each of the following questions with **T** or **F** and **provide a one line justification**.

(a) [2 pts.] Consider two datasets $D^{(1)}$ and $D^{(2)}$ where $D^{(1)} = \{(x_1^{(1)}, y_1^{(1)}), ..., (x_n^{(1)}, y_n^{(1)})\}$ and $D^{(2)} = \{(x_1^{(2)}, y_1^{(2)}), ..., (x_m^{(2)}, y_m^{(2)})\}$ such that $x_i^{(1)} \in \mathbb{R}^{d_1}$, $x_i^{(2)} \in \mathbb{R}^{d_2}$. Suppose $d_1 > d_2$ and $n > m$. Then the maximum number of mistakes a perceptron algorithm will make is higher on dataset $D^{(1)}$ than on dataset $D^{(2)}$.

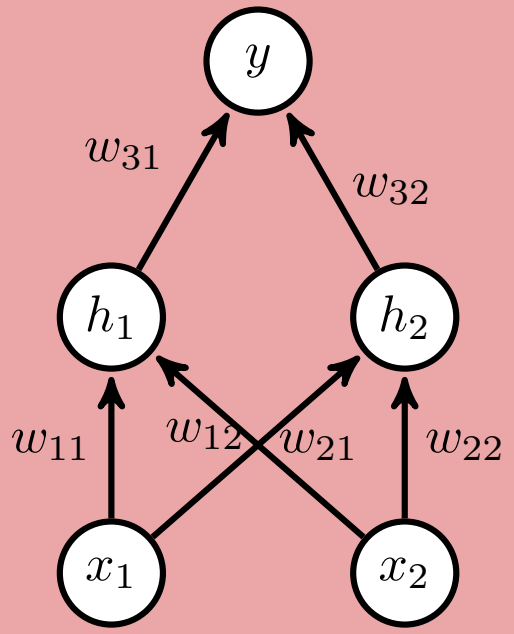
Sample Questions

Neural Networks

Can the neural network in Figure (b) correctly classify the dataset given in Figure (a)?



(a) The dataset with groups S_1, S_2, and S_3.

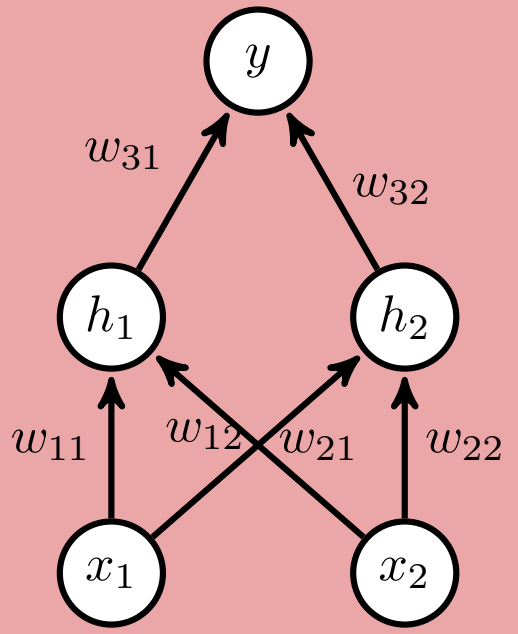


(b) The neural network architecture

Sample Questions

Neural Networks

Apply the backpropagation algorithm to obtain the partial derivative of the mean-squared error of y with the true value y^* with respect to the weight w_{22} assuming a sigmoid nonlinear activation function for the hidden layer.



(b) The neural network architecture

The Big Picture

CLASSIFICATION AND REGRESSION

Classification and Regression: The Big Picture

Whiteboard

- Decision Rules / Models
- Objective Functions
- Regularization
- Update Rules
- Nonlinear Features

Q&A

GENERALIZATION IN ML

ML Big Picture

Learning Paradigms:

What data is available and when? What form of prediction?

- supervised learning
- unsupervised learning
- semi-supervised learning
- reinforcement learning
- active learning
- imitation learning
- domain adaptation
- online learning
- density estimation
- recommender systems
- feature learning
- manifold learning
- dimensionality reduction
- ensemble learning
- distant supervision
- hyperparameter optimization

Theoretical Foundations:

What principles guide learning?

- ❑ probabilistic
- ❑ information theoretic
- ❑ evolutionary search
- ❑ ML as optimization

Problem Formulation:

What is the structure of our output prediction?

boolean	Binary Classification
categorical	Multiclass Classification
ordinal	Ordinal Classification
real	Regression
ordering	Ranking
multiple discrete	Structured Prediction
multiple continuous	(e.g. dynamical systems)
both discrete & cont.	(e.g. mixed graphical models)

Application Areas

Key challenges?

NLP, Speech, Computer Vision, Robotics, Medicine, Search

Facets of Building ML Systems:

How to build systems that are robust, efficient, adaptive, effective?

1. *Data prep*
2. *Model selection*
3. *Training (optimization / search)*
4. *Hyperparameter tuning on validation data*
5. *(Blind) Assessment on test data*

Big Ideas in ML:

Which are the ideas driving development of the field?

- *inductive bias*
- *generalization / overfitting*
- *bias-variance decomposition*
- *generative vs. discriminative*
- *deep nets, graphical models*
- *PAC learning*
- *distant rewards*

LEARNING THEORY

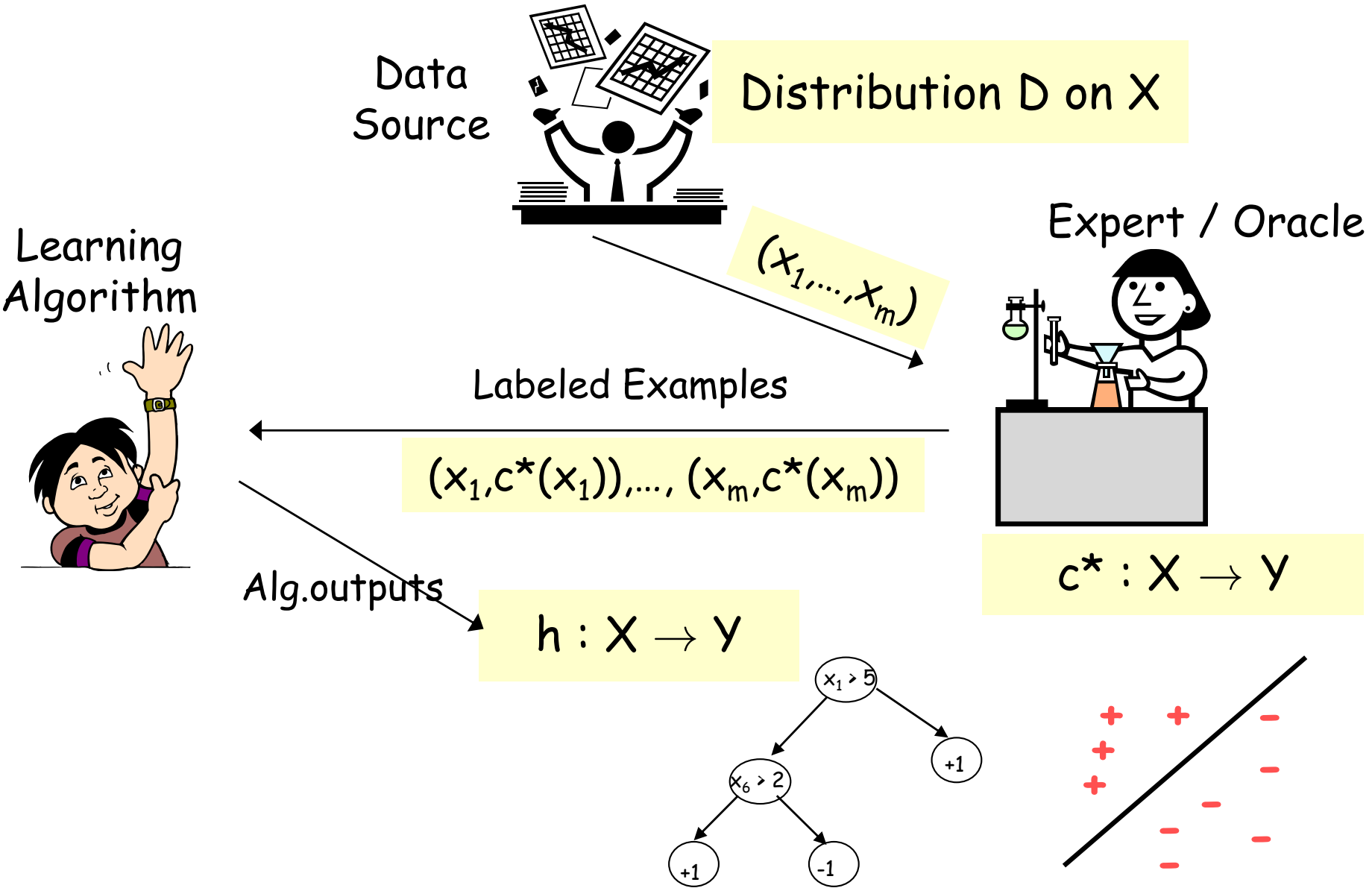
Questions For Today

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

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

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

PAC/SLT models for Supervised Learning



Slide from Nina Balcan

Two Types of Error

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**

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

We've also referred to this as the "Function Approximation View"

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:

$$h^* = \underset{h \in \mathcal{H}}{\operatorname{argmin}} R(h) \tag{2}$$

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

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

PAC LEARNING

Probably Approximately Correct (PAC) Learning

Whiteboard:

- PAC Criterion
- Meaning of "Probably Approximately Correct"
- PAC Learnable
- Consistent Learner
- Sample Complexity

Generalization and Overfitting

Whiteboard:

- Realizable vs. Agnostic Cases
- Finite vs. Infinite Hypothesis Spaces

PAC Learning

The **PAC criterion** is that our learner produces a high accuracy learner with high probability:

$$P(|R(h) - \hat{R}(h)| \leq \epsilon) \geq 1 - \delta \tag{1}$$

Suppose we have a learner that produces a hypothesis $h \in \mathcal{H}$ given a sample of N training examples. The algorithm is called **consistent** if for every ϵ and δ, there exists a positive number of training examples N such that for any distribution p^*, we have that:

$$P(|R(h) - \hat{R}(h)| > \epsilon) < \delta \tag{2}$$

The **sample complexity** is the minimum value of N for which this statement holds. If N is finite for some learning algorithm, then $\mathcal{H}$ is said to be **learnable**. If N is a polynomial function of $\frac{1}{\epsilon}$ and $\frac{1}{\delta}$ for some learning algorithm, then $\mathcal{H}$ is said to be **PAC learnable**.

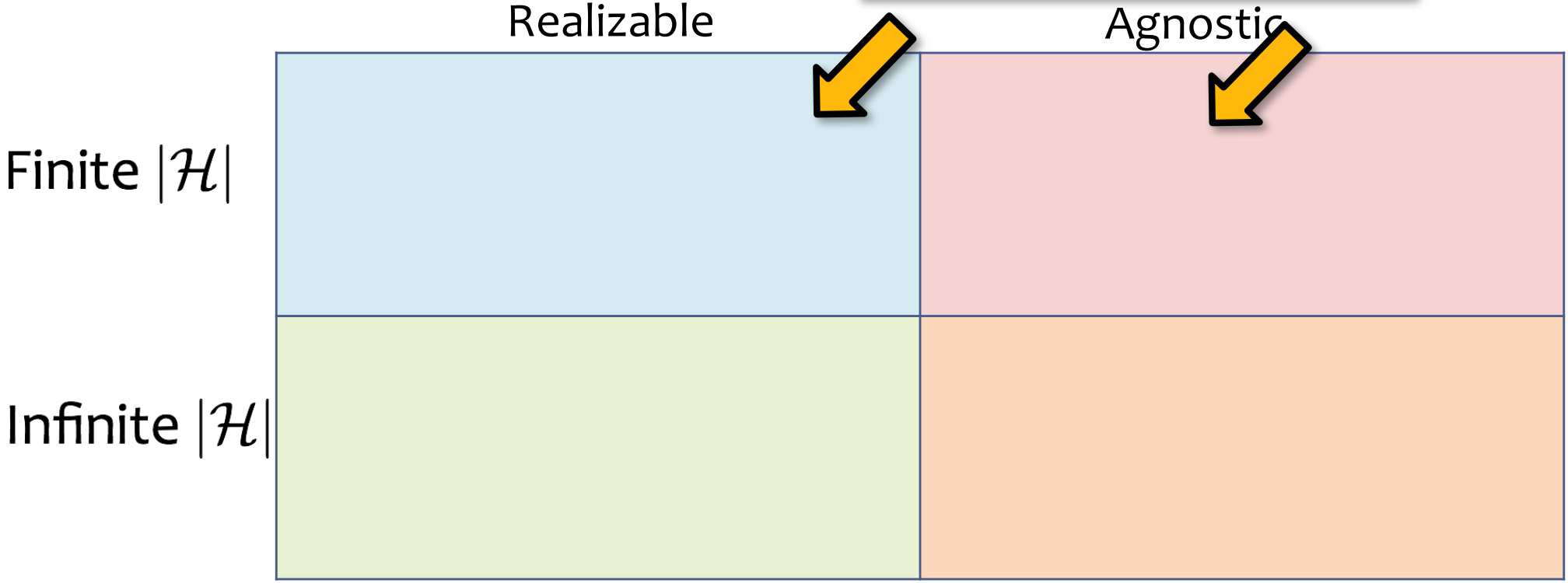
SAMPLE COMPLEXITY RESULTS

Sample Complexity Results

Definition 0.1. The **sample complexity** of a learning algorithm is the number of examples required to achieve arbitrarily small error (with respect to the optimal hypothesis) with high probability (i.e. close to 1).

Four Cases we care about…

We'll start with the finite case…



Sample Complexity Results

Definition 0.1. The **sample complexity** of a learning algorithm is the number of examples required to achieve arbitrarily small error (with respect to the optimal hypothesis) with high probability (i.e. close to 1).

Four Cases we care about…

	Realizable	Agnostic
Finite $\lvert\mathcal{H}\rvert$	$N \geq \frac{1}{\epsilon}\left[\log(\lvert\mathcal{H}\rvert) + \log(\frac{1}{\delta})\right]$ labeled examples are sufficient so that with probability $(1-\delta)$ all $h \in \mathcal{H}$ with $R(h) \geq \epsilon$ have $\hat{R}(h) > 0$.	
Infinite $\lvert\mathcal{H}\rvert$		

Example: Conjunctions

In-Class Quiz:

Suppose H = class of conjunctions over **x** in $\{0,1\}^M$

If M = 10, ε = 0.1, δ = 0.01, how many examples suffice?

	Realizable	Agnostic
Finite $\lvert\mathcal{H}\rvert$	$N \geq \frac{1}{\epsilon}\left[\log(\lvert\mathcal{H}\rvert) + \log(\frac{1}{\delta})\right]$ labeled examples are sufficient so that with probability $(1-\delta)$ all $h \in \mathcal{H}$ with $R(h) \geq \epsilon$ have $\hat{R}(h) > 0$.	
Infinite $\lvert\mathcal{H}\rvert$		

Sample Complexity Results

Definition 0.1. The **sample complexity** of a learning algorithm is the number of examples required to achieve arbitrarily small error (with respect to the optimal hypothesis) with high probability (i.e. close to 1).

Four Cases we care about…

	Realizable	Agnostic
Finite $\lvert\mathcal{H}\rvert$	$N \geq \frac{1}{\epsilon}\left[\log(\lvert\mathcal{H}\rvert) + \log(\frac{1}{\delta})\right]$ labeled examples are sufficient so that with probability $(1-\delta)$ all $h \in \mathcal{H}$ with $R(h) \geq \epsilon$ have $\hat{R}(h) > 0$.	$N \geq \frac{1}{2\epsilon^2}\left[\log(\lvert\mathcal{H}\rvert) + \log(\frac{2}{\delta})\right]$ labeled examples are sufficient so that with probability $(1-\delta)$ for all $h \in \mathcal{H}$ we have that $\lvert R(h) - \hat{R}(h)\rvert < \epsilon$.
Infinite $\lvert\mathcal{H}\rvert$		

1. Bound is **inversely linear in epsilon** (e.g. halving the error requires double the examples)
2. Bound is **only logarithmic in |H|** (e.g. quadrupling the hypothesis space only requires double the examples)

1. Bound is **inversely quadratic in epsilon** (e.g. halving the error requires 4x the examples)
2. Bound is **only logarithmic in |H|** (i.e. same as Realizable case)

	Realizable	Agnostic
Finite $\lvert\mathcal{H}\rvert$	$N \geq \frac{1}{\epsilon}\left[\log(\lvert\mathcal{H}\rvert) + \log(\frac{1}{\delta})\right]$ labeled examples are sufficient so that with probability $(1-\delta)$ all $h \in \mathcal{H}$ with $R(h) \geq \epsilon$ have $\hat{R}(h) > 0$.	$N \geq \frac{1}{2\epsilon^2}\left[\log(\lvert\mathcal{H}\rvert) + \log(\frac{2}{\delta})\right]$ labeled examples are sufficient so that with probability $(1-\delta)$ for all $h \in \mathcal{H}$ we have that $\lvert R(h) - \hat{R}(h)\rvert < \epsilon$.
Infinite $\lvert\mathcal{H}\rvert$		

Generalization and Overfitting

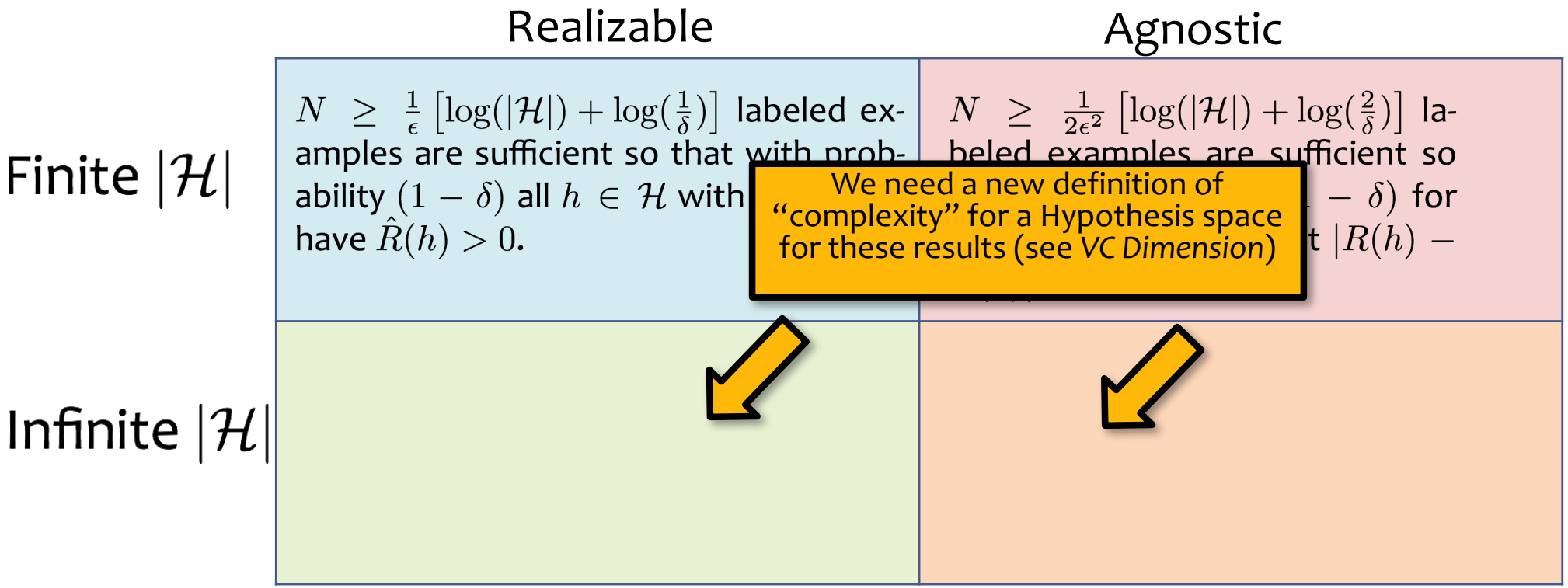
Whiteboard:

- Sample Complexity Bounds (Agnostic Case)
- Corollary (Agnostic Case)
- Empirical Risk Minimization
- Structural Risk Minimization
- Motivation for Regularization

Sample Complexity Results

Definition 0.1. The **sample complexity** of a learning algorithm is the number of examples required to achieve arbitrarily small error (with respect to the optimal hypothesis) with high probability (i.e. close to 1).

Four Cases we care about…



Sample Complexity Results

Definition 0.1. The **sample complexity** of a learning algorithm is the number of examples required to achieve arbitrarily small error (with respect to the optimal hypothesis) with high probability (i.e. close to 1).

Four Cases we care about…

	Realizable	Agnostic
Finite $\lvert\mathcal{H}\rvert$	$N \geq \frac{1}{\epsilon}\left[\log(\lvert\mathcal{H}\rvert) + \log(\frac{1}{\delta})\right]$ labeled examples are sufficient so that with probability $(1-\delta)$ all $h \in \mathcal{H}$ with $R(h) \geq \epsilon$ have $\hat{R}(h) > 0$.	$N \geq \frac{1}{2\epsilon^2}\left[\log(\lvert\mathcal{H}\rvert) + \log(\frac{2}{\delta})\right]$ labeled examples are sufficient so that with probability $(1-\delta)$ for all $h \in \mathcal{H}$ we have that $\lvert R(h) - \hat{R}(h)\rvert < \epsilon$.
Infinite $\lvert\mathcal{H}\rvert$	$N = O(\frac{1}{\epsilon}\left[\text{VC}(\mathcal{H})\log(\frac{1}{\epsilon}) + \log(\frac{1}{\delta})\right])$ labeled examples are sufficient so that with probability $(1-\delta)$ all $h \in \mathcal{H}$ with $R(h) \geq \epsilon$ have $\hat{R}(h) > 0$.	$N = O(\frac{1}{\epsilon^2}\left[\text{VC}(\mathcal{H}) + \log(\frac{1}{\delta})\right])$ labeled examples are sufficient so that with probability $(1-\delta)$ for all $h \in \mathcal{H}$ we have that $\lvert R(h) - \hat{R}(h)\rvert \leq \epsilon$.

Generalization and Inductive Bias

Chalkboard:

- Setting: binary classification with binary feature vectors
- Instance space vs. Hypothesis space
- Counting: # of instances, # leaves in a full decision tree, # of full decision trees, # of labelings of training examples
- Algorithm: keep all full decision trees consistent with the training data and do a majority vote to classify
- Case study: training size is all, all-but-one, all-but-two, all-but-three,…