CS 158

Machine Learning

Instructor: Jessica Wu

The instructor gratefully acknowledges Eric Eaton (UPenn), who assembled the original slides, David Kauchak (Pomona), whose slides are also heavily used, and the many others who made their course materials freely available online.

ML Basics

Learning Goals

Learning Goals

- Define Machine Learning
- Know when ML is useful

Machine Learning is…

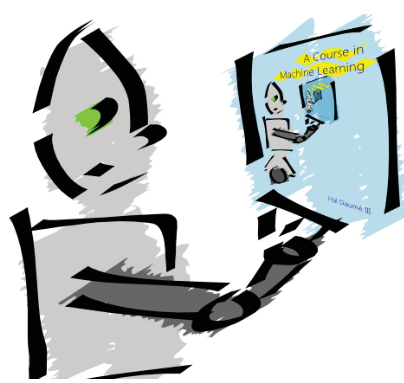
Machine learning, a branch of artificial intelligence, concerns the construction and study of systems that can learn from data.

Slide credit: David Kauchak

Machine Learning is…

Machine learning is about predicting the future based on the past.

-- Hal Daume III



Slide credit: David Kauchak

Machine learning is…

Machine Learning is the study of algorithms that

- improve their performance P
- at some task T
- with experience E.

A well-defined learning task is given by <P, T, E>.

[Definition by Tom Mitchell (1998)]

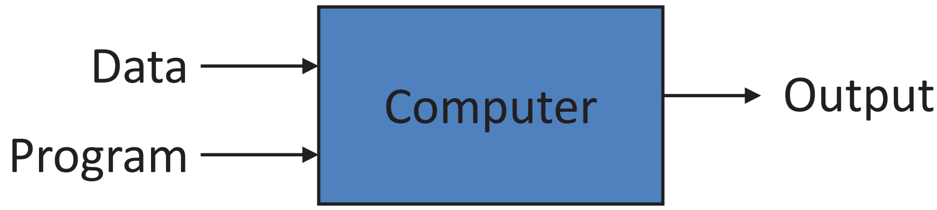
Based on slide by Eric Eaton

Samuel's Checkers-Player

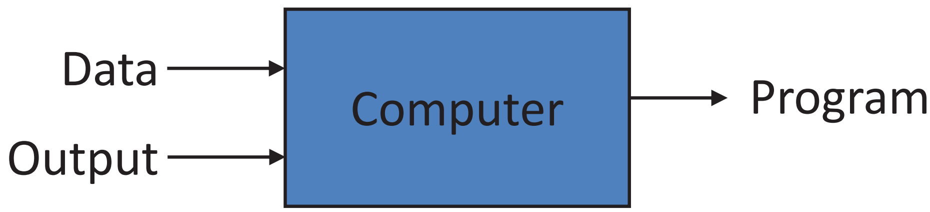
"Machine Learning: Field of study that gives computers the ability to learn without being explicitly programmed." -Arthur Samuel (1959)

Slide credit: Eric Eaton

Traditional Programming



Machine Learning



Slide credit: Pedro Domingos

Defining the Learning Task

Improve on task T, with respect to performance metric P, based on experience E

T: Playing checkers

T: Recognizing hand-written words

T: Driving on four-lane highways using vision sensors

T: Categorize email messages as spam or legitimate

(indicates you need to fill in slide)

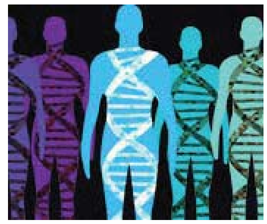
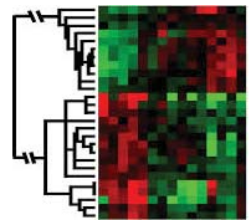
Slide credit: Ray Mooney

When Do We Use Machine Learning?

ML is used when:

- Human expertise does not exist (navigating on Mars)
- Humans cannot explain their expertise (speech recognition)
- Models must be customized (personalized medicine)
- Models are based on huge amounts of data (genomics)



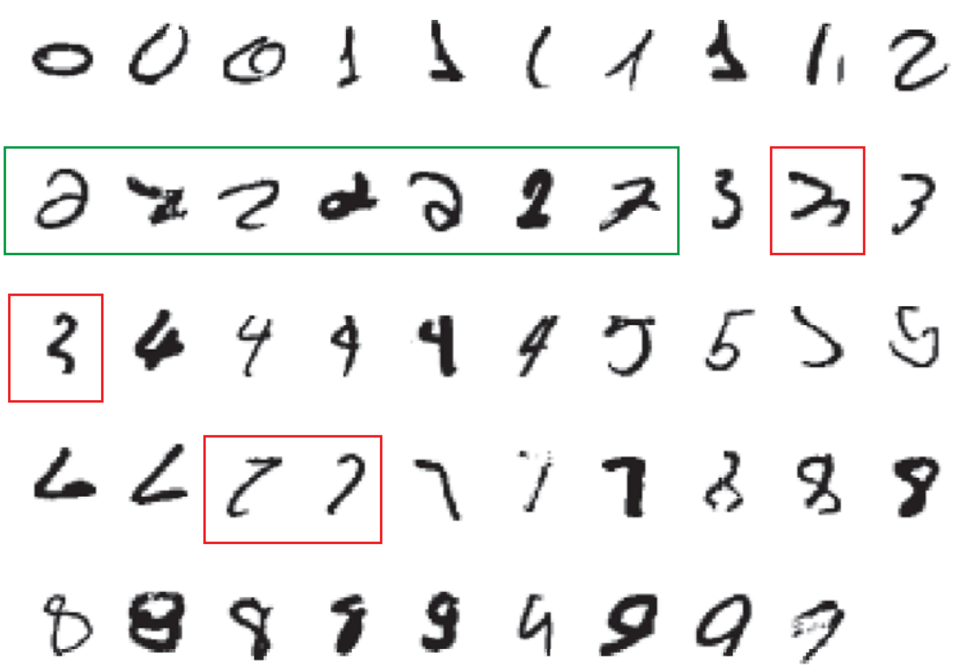



Learning is not always useful:

- There is no need to "learn" to calculate payroll

Slide credit: Eric Eaton
Based on slide by E. Alpaydin

A classic example of a task that requires machine learning:
It is very hard to say what makes a 2



Slide credit: Geoffrey Hinton

Some examples of tasks that are best solved by using a learning algorithm

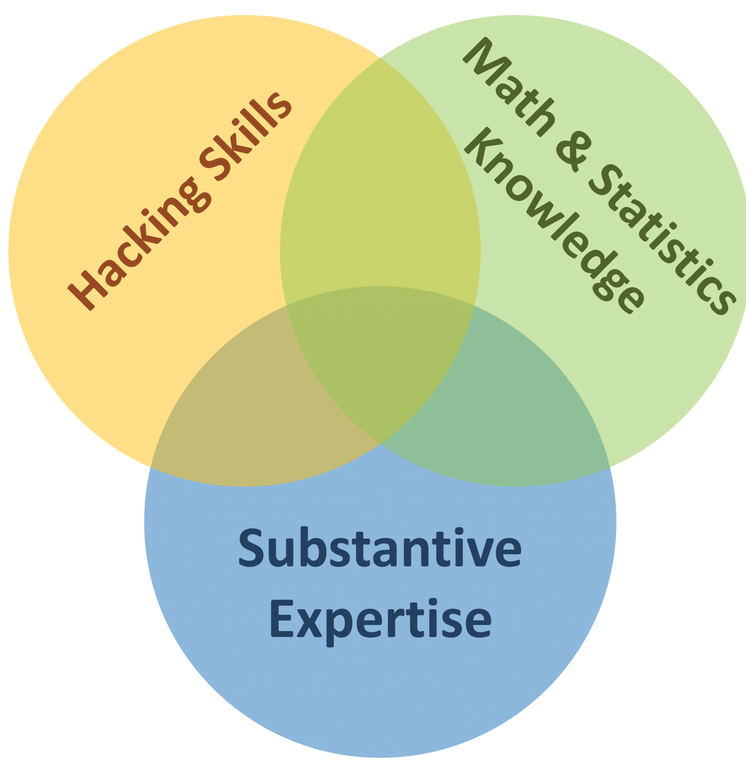
- Learning patterns
  - Predicting passenger survivability on the *Titanic*
  - Recognizing tweets as positive or negative
  - Clustering faces by identity
- State-of-the-art applications
  - Autonomous cars
  - Automatic speech recognition
  - Anomalies in credit card transactions
  - Stock prediction
- [Your favorite area here]

(This slide intentionally left blank.)

Course Basics

Learning Goals

- Know course goals
- Know course policies



Drew Conway's Data Science Venn Diagram

How much Math?

- A Fall 2015 course project analyzed syllabi text of all 5C courses and found that CS 158 is ...
  - 46.4% Math (36.4% theory + 10.0% linear systems + applications)
  - 13.6% CS (engineering, design, programming)
  - 10.7% lab sciences (research, field, analysis)
  - 10.0% research (thesis, project, writing)
  - 7.9% social science
  - 5.0% humanities
  - 6.4% other
- What can you do?
  - Evaluate your background: PS 1 has math refresher and resources
  - Let me know when math is going too fast!

What have Students Said?

"The course ... is well-balanced in terms of math and theory (coding) parts. The course helped me review many important math theory and techniques."

"I liked the balance between implementing algorithms and using existing libraries."

"I enjoyed the amount of coding for the homeworks / projects and think that it went well with the theory we learned in class."

"I liked the real-world applications of ML that we learned about."

Types of Learning

Learning Goals

- Define three main types of machine learning
- List goals and inputs and outputs

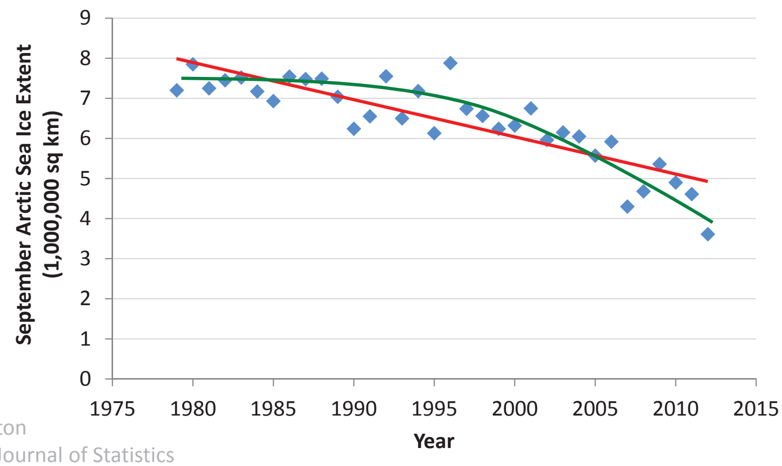
Types of Learning

- **Supervised (inductive) learning :** Learn with a teacher
  - Given: labeled training examples
  - Goal: learn mapping that predicts label for test example
- **Unsupervised learning :** Learn without a teacher
  - Given: unlabeled inputs
  - Goal: learn some intrinsic structure in inputs
- **Reinforcement learning:** Learn by interacting
  - Given agent interacting in environment (having set of states)
  - Learn policy (state to action mapping) that maximizes agent's reward

Slide credit: Eric Eaton
Based on slide by Pedro Domingos

Supervised Learning: Regression

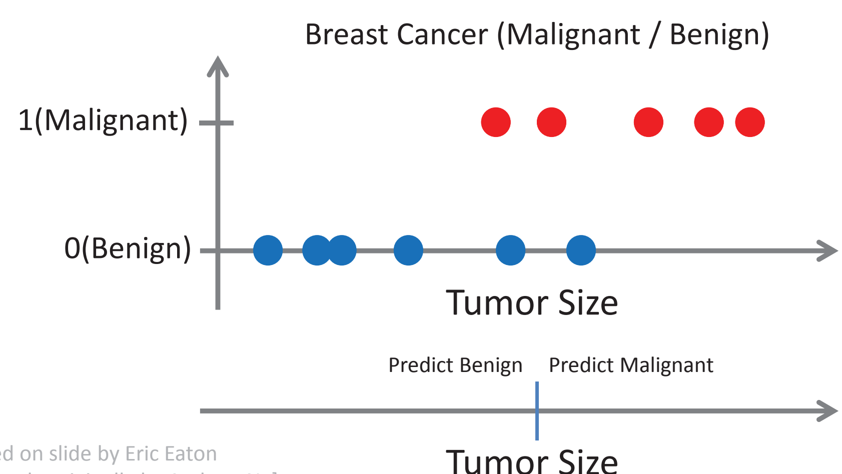
- Given $(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \ldots, (x^{(n)}, y^{(n)})$
- Learn a function $f(x)$ to predict y given x
  - y is real-valued == regression



Slide credit: Eric Eaton
Data from G. Witt. Journal of Statistics Education, Volume 21, Number 1 (2013)

Supervised Learning: Classification

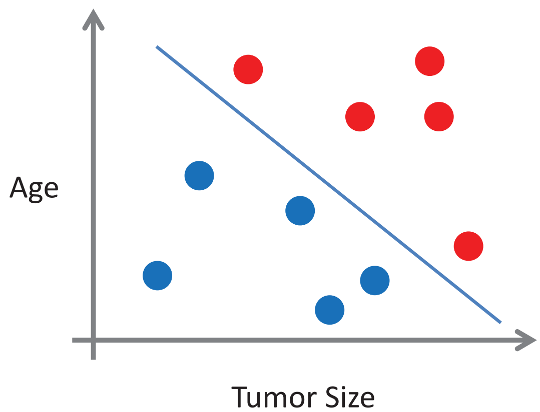
- Given $(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \ldots, (x^{(n)}, y^{(n)})$
- Learn a function $f(x)$ to predict y given x
  - y is categorical == classification



Based on slide by Eric Eaton
[example originally by Andrew Ng]

Supervised Learning

- x can be multi-dimensional
  - each dimension corresponds to an attribute



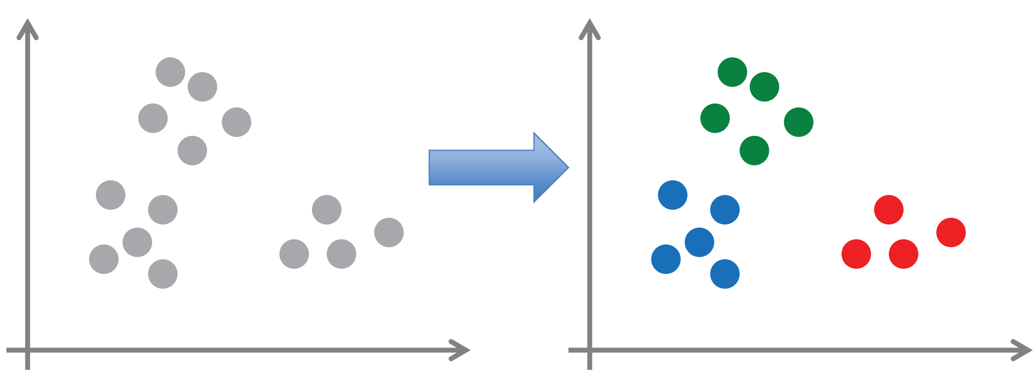
- Clump Thickness
- Uniformity of Cell Size
- Uniformity of Cell Shape

…

Based on slide by Eric Eaton
[example originally by Andrew Ng]

Unsupervised Learning

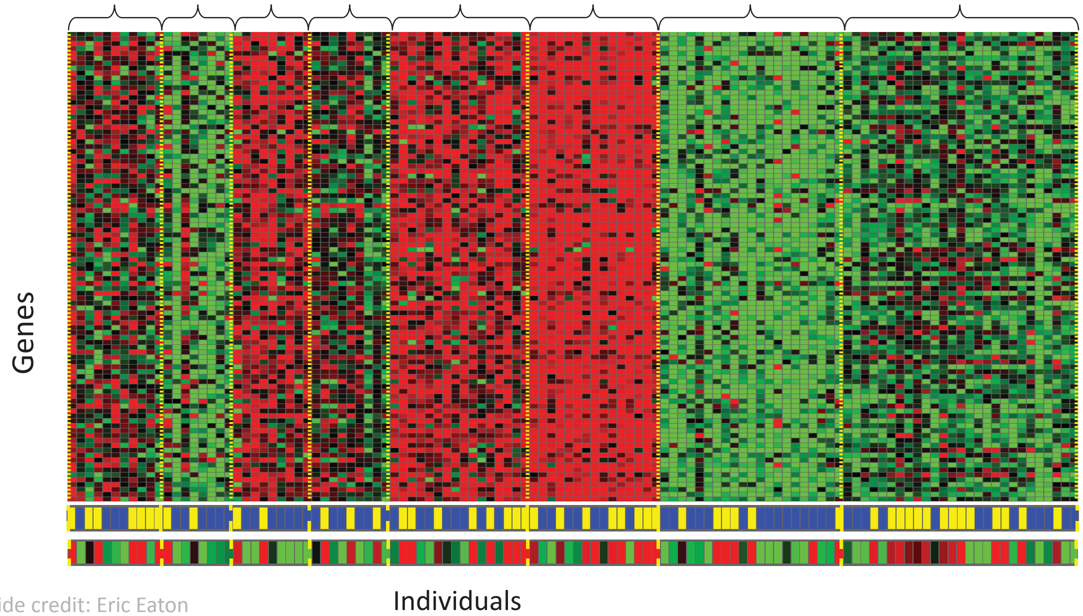
- Given $x^{(1)}, x^{(2)}, \ldots, x^{(n)}$ (without labels)
- Output hidden structure behind the x's
  - e.g., clustering



Slide credit: Eric Eaton

Unsupervised Learning

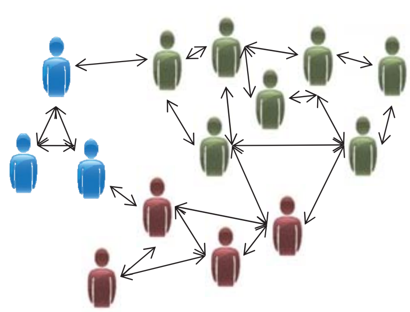
Genomics application: group individuals by genetic similarity



Slide credit: Eric Eaton
[Source: Daphne Koller]

Unsupervised Learning

Market segmentation



Social network analysis

Based on slide by Andrew Ng

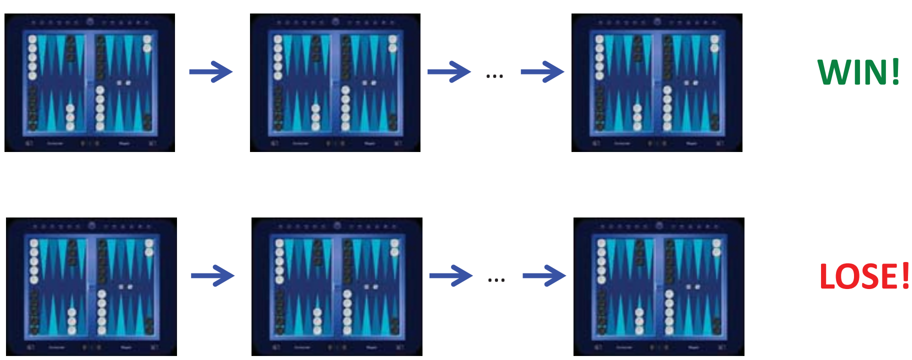
Reinforcement Learning

- Given sequence of states and actions with (delayed) rewards
- Learn policy that maximizes agent’s reward
- Examples:
  – Credit assignment problem
  – Game playing
  – Robot in maze
  – Balance a pole on your hand

Slide credit: Eric Eaton

Reinforcement Learning

Backgammon



Given sequences of moves and whether or not the player won at the end, learn to make good moves

Slide credit: David Kauchak

Reinforcement Learning

https://www.youtube.com/watch?v=4cgWya-wjgY

Slide credit: Eric Eaton

Brief Peek at Topics

- **Supervised Learning**
  - Decision trees
  - K-nearest neighbors
  - Linear regression
  - Logistic regression
  - Perceptron
  - Support vector machines
  - Ensemble methods
- **Unsupervised Learning**
  - Principal component analysis
  - Clustering
  - Expectation maximization
- **Reinforcement Learning**
  - None: covered in-depth in AI (CS 151)
- **Structured Learning**
  - Hidden Markov Models
  - Collaborative filtering
- **Other Topics**
  - Experimental evaluation (procedure + metrics)
  - Computational learning theory

Framing a Learning Problem

Learning Goals

- List stages of ML pipeline
- Name some key issues in ML

Representing Examples

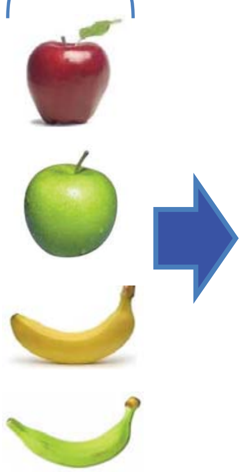
What is an example?

How is it represented?

examples

features

$feat_1, feat_2, feat_3, feat_4, \ldots$



How our algorithms actually “view” the data

Features are the questions we can ask about the examples

Based on slide by David Kauchak

Learning System

examples | label

red, round, leaf, 3oz, … apple

green, round, no leaf, 4oz, … apple

yellow, curved, no leaf, 4oz, … banana

green, curved, no leaf, 5oz, … banana

learn

model/ classifier

During learning/training/induction, learn a model of what distinguishes apples and bananas *based on the features*

red, round, no leaf, 4oz, …

model/ classifier

The model can then classify a new example *based on the features*

Based on slide by David Kauchak

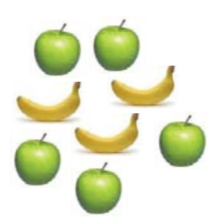
Learning System

- Learning is about ***generalizing*** from training data
- What does this ***assume*** about training and test set?

Training data

Test set

Training data



Test set

Not always the case, but we'll often assume it is!

Based on slide by David Kauchak

More Technically…

- We are going to use the ***probabilistic model*** of learning
- There is some (unknown) probability distribution over example/label pairs called the ***data generating distribution***
- **Both** the training data **and** the test set are generated based on this distribution
  - we call this i.i.d. (independent and identically distributed)

Based on slide by David Kauchak

ML as Function Approximation

Problem Setting

- Set of possible instances $\mathcal{X}$
- Set of possible labels $\mathcal{Y}$
- Unknown target function $f : \mathcal{X} \to \mathcal{Y}$
- Set of function hypotheses $H = \{h \mid h : \mathcal{X} \to \mathcal{Y}\}$

Input: Training examples of unknown target function f

$$\left\{\left(x^{(i)}, y^{(i)}\right)\right\}_{i=1}^{n} = \left\{\left(x^{(1)}, y^{(1)}\right), \ldots, \left(x^{(n)}, y^{(n)}\right)\right\}$$

(superscript denotes example number)

Output: Hypothesis $h \in H$ that best approximates f

Slide credit: Eric Eaton
Based on slide by Tom Mitchell

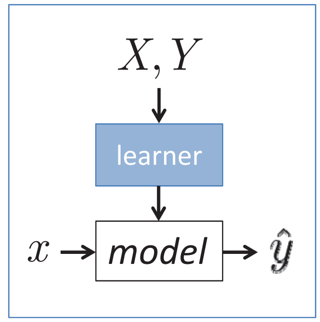
Stages of (Batch) Machine Learning

Given: labeled training data $X, Y = \left\{ \left(x^{(i)}, y^{(i)} \right) \right\}_{i=1}^{n}$

- Assumes each $(x^{(i)}, y^{(i)}) \sim p(x, y)$ such that $y^{(i)} = f(x^{(i)})$

Train model:

model ← *learner*.train(X, Y)



Apply model to new data:

- Given: new unlabeled instance $x \sim p(x)$

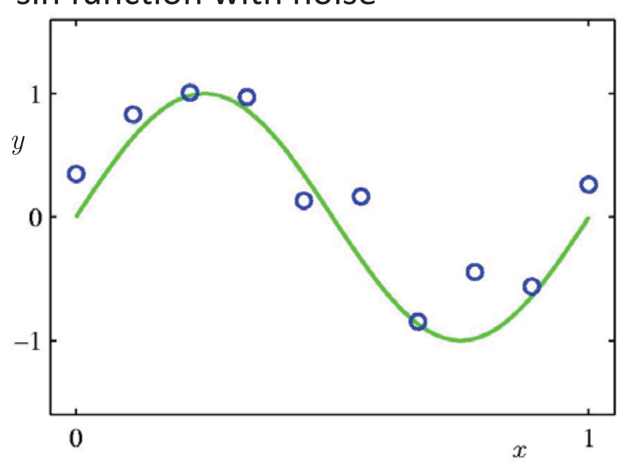
$\hat{y}$ ← *model*.predict(x)

Based on slide by Eric Eaton

Example Regression Problem

- Consider simple regression dataset
  - $f : \mathcal{X} \rightarrow \mathcal{Y}$
  - $x \in \mathbb{R}$
  - $y \in \mathbb{R}$

Dataset: 10 points generated from sin function with noise

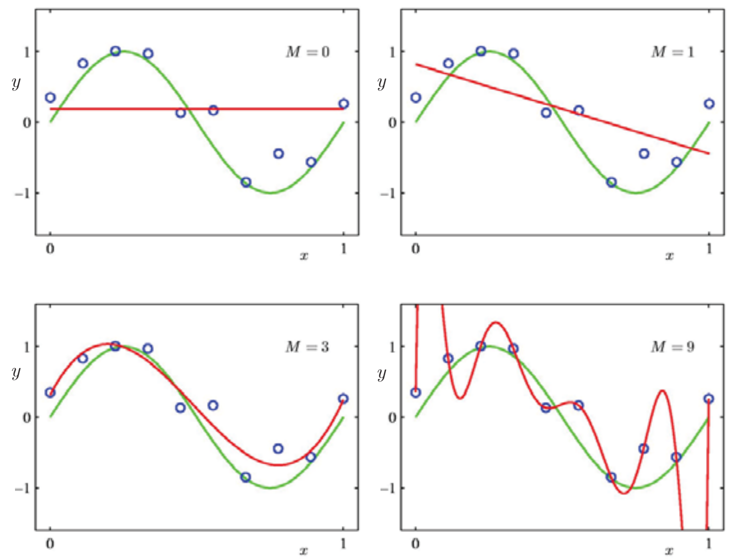


- Question 1: How should we pick the hypothesis space H?
- Question 2: How do we find the best h in this space?

Based on slide by David Sontag
Images from Bishop [PRML]

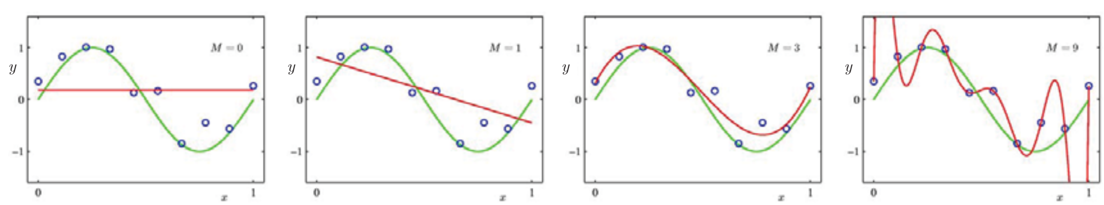
Hypothesis Space: Degree-M Polynomials

- Infinitely many hypotheses
- *M* < 9 not consistent with dataset, *M* ≥ 9 consistent
- Which one is **best**?



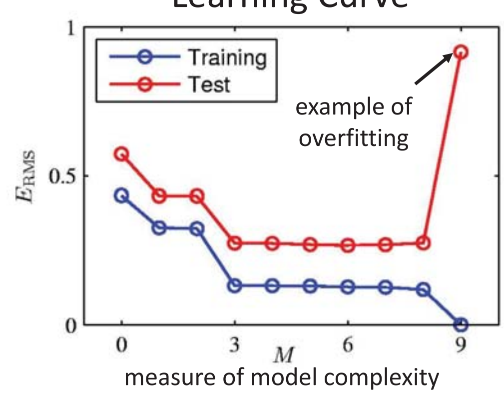
Based on slide by David Sontag
Images from Bishop [PRML]

Hypothesis Space: Degree-M Polynomials



- We measure error using a loss function $L(y, \hat{y})$
- For regression, common choice is squared loss $$L\left(y^{(i)}, h(x^{(i)})\right) = \left(y^{(i)} - h(x^{(i)})\right)^2$$
- *Empirical loss* of function h applied to training data is then $$\frac{1}{n}\sum_{i=1}^{n} L\left(y^{(i)}, h(x^{(i)})\right) = \frac{1}{n}\sum_{i=1}^{n}\left(y^{(i)} - h(x^{(i)})\right)^2$$

Learning Curve



Based on slide by David Sontag
Images from Bishop [PRML]

Key Issues in Machine Learning

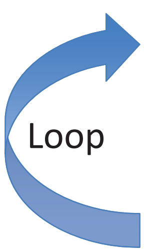
Representation : How do we choose a hypothesis space?
- Often we use **prior knowledge** to guide this choice
- The ability to answer the next two questions also affects choice

Optimization : How do we find the best hypothesis within this space?
- This is an **algorithmic** question, at the intersection of computer science and optimization research.

Evaluation : How can we gauge the accuracy of a hypothesis on unseen testing data?
- The previous example showed that choosing the hypothesis which simply minimizes training set error **is not optimal**
- This question is the main topic of **learning theory**

Based on slides by Eric Eaton and by David Sontag

ML in Practice



- Understand domain, prior knowledge, and goals
- Data integration, selection, cleaning, pre-processing, etc.
- Learn models
- Interpret results
- Consolidate and deploy discovered knowledge

Lessons Learned about Learning

- Learning can be viewed as using experience to approximate a chosen target function
- Function approximation can be viewed as a search through a space of hypotheses (representations of functions) for one that best fits a set of training data
- Different learning methods assume different hypothesis spaces and/or employ different search techniques

Slide credit: Eric Eaton and Ray Mooney
Based on a slide by Pedro Domingos