CMSC 726
Lecture 2:ML Basics

Lise Getoor

September 2, 2010

ACKNOWLEDGEMENTS: The material in this course is a synthesis of materials from many sources, including: Hal Daume III, Mark Drezde, Carlos Guestrin, Andrew Ng, Ben Taskar, Eric Xing, and others. I am very grateful for their generous sharing of insights and materials.

Today's Topics

- ML: Motivating Applications
- ML Paradigms
- Model Selection
- Evaluation of ML Algorithms
- The Curse of Dimensionality

Machine Learning

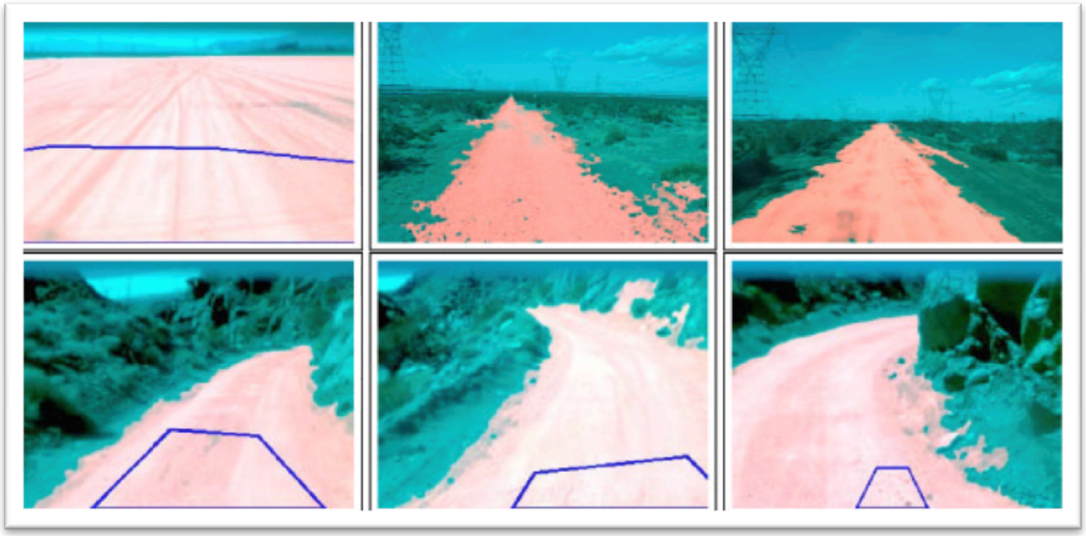
» Applications

Computational Biology

- Process, understand, and organize biological data
- Tasks:
  - Gene finding
  - Classification of gene expression data
  - Inferring regulatory networks from mRNA and proteomic data
- Methods
  - Graphical models
  - HMMs
  - SVMs
  - Etc.

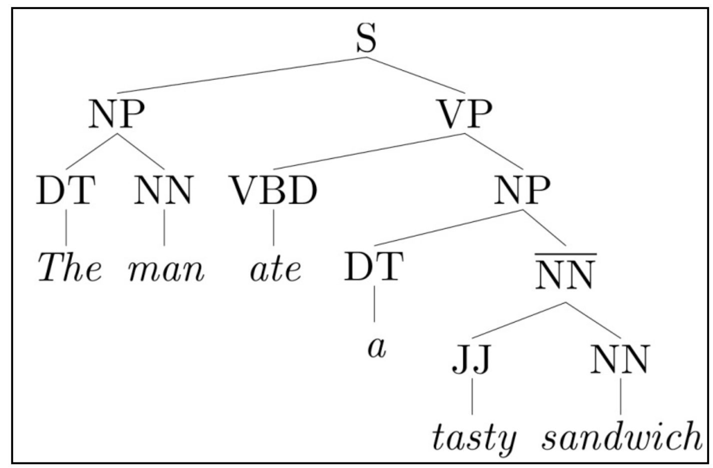
Computer Vision

- Process, understand, and organize images to identify objects and patterns
- Tasks:
  - Face recognition
  - Scene Understanding
- Methods:
  - Graphical models
  - Dimensionality reduction



Natural Language Processing

- Process, understand and organize language and text data
- Tasks:
  - Machine translation
  - Web search
  - Spam classification
- Methods
  - Linear classification
  - Graphical models
  - HMMs
  - Structured learning



Robotics

- Process, understand, and organize sensor and other data to achieve goals
- Involves:
  - Planning
    - How to achieve goals, where to move
  - Sensors/Vision
    - Identifying objects, analyzing properties
  - Haptics
    - Control of movement
- Examples
  - Helicopter control: http://heli.stanford.edu/
- Methods
  - Reinforcement learning
  - Classification
  - Dimensionality reduction

YouTube Video.

Machine Learning in 3 Parts

Input/Output

Representation

Algorithm

What is learning?
How do we learn?

Evaluation

Evaluation

Input & Output

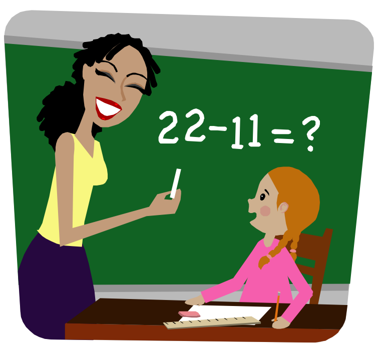
» Settings and Representations

Learning Settings

- What information is available?
  - Types of data
  - Annotations
  - Examples of output
- What output is desired?
  - How will the algorithm be used?

Supervised Learning

- Learning with a teacher
  - Explicit feedback in the form of labeled examples
  - Goal: Make prediction
  - Pros: Often can get good performance
  - Cons: Labeled data is sometimes difficult to find
- Examples
  - Classification
    - Sort documents by topic
  - Regression
    - Stock price
  - Structured prediction
    - Identify parts of an image
  - Ranking
    - Sort web pages



Unsupervised Learning

- Learning by oneself
  - Only observed unlabeled examples
  - Goal: uncover structure in data
  - Pros: Easy to find lots of data
  - Cons: Finding patterns of interest
- Examples
  - Clustering
    - Group emails by topic
  - Manifold learning
    - Find a low dimensional data representation
  - Segmentation
    - Divide video into scenes

Semi-Supervised Learning

- Some labeled examples + lots of unlabeled examples
- Lots of ways to do this
  - Use unlabeled to guide learning in classification
    - Some documents labeled by topic, lots of unsorted docs
  - Graph-based models for labeling new data
    - Label propagation
  - Other weak forms of supervision
    - A list of names, learn to extract more

Reinforcement Learning

- Learn a behavior policy by interacting with the world
  - How to navigate in a world
  - Success measured by rewards received by actions
  - Maximize rewards – costs
- No examples
  - You don't know how you did till its over
    - Ex. Chess– was that a good move?
      Did you eventually win?
- Examples
  - Chess (and checkers) and game agents
  - Robot control

How Do We Represent Data?

- Data is complex
- How does a learning algorithm see data?

High Dimensional Vectors

- A learning example is a vector of length M

$$x_i \in \Re^M$$

- Each dimension represents a feature
  - Feature functions f_j

- Dataset: A collection of N examples

$$D = \{x_i\}_{i=1}^N$$

- Columns are the feature functions

$$x[j] = f_j(D)$$

What Does Data Look Like?

- Predict airplane delay

Label	Bad weather	Other planes delayed	Ontime performance of route	
Yes	Yes	5	60%	x_1
No	Yes	0	95%	x_2
Yes	No	2	70%	x_3
No	No	1	80%	x_4
y	$x[1]$	$x[2]$	$x[3]$	

Categorical/binary feature   Numeric feature

Ordinal features: e.g. (small, med, large)

What Does Data Look Like?

Face recognition Input: faces Predict: does picture have face?

Representations

x

- Designing feature functions is critical
  - Well designed representations greatly effect performance
- How should you design features?
  - Features are application specific
  - You need to know about biology/vision/speech/etc.
- Since this is domain specific we won't talk much about it

Which Learning Setting?

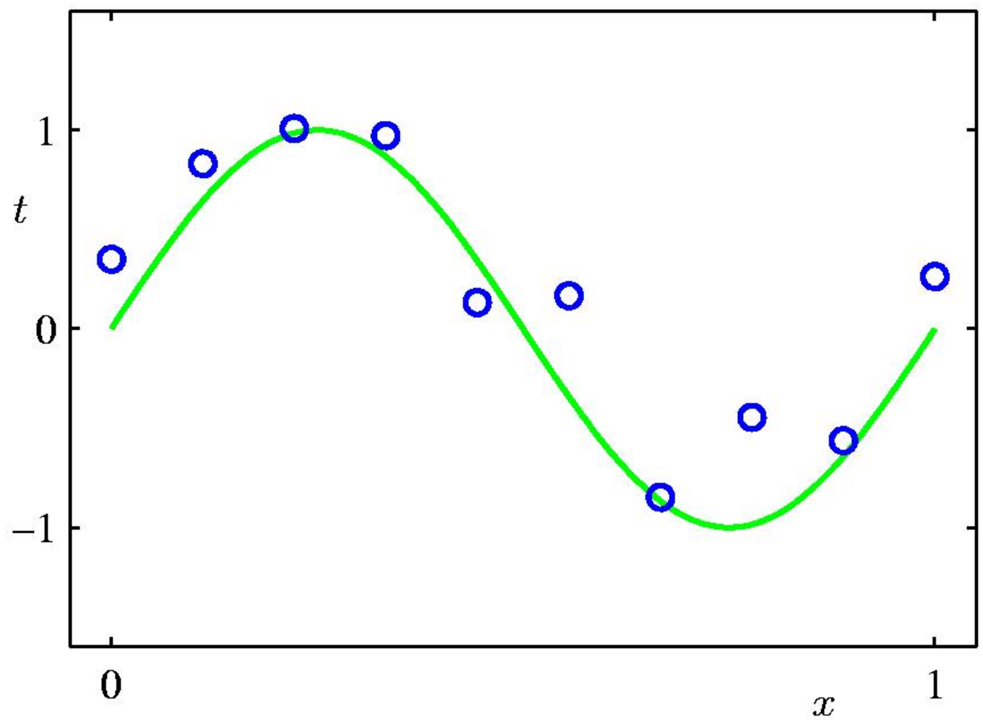
- Email users can mark messages as spam. We want to filter these messages before the user ever sees them.

- Our lab collected gene expression profiles by subjecting each gene to different stimuli. Using these profiles we want to group genes by similar expression patterns.

Algorithm

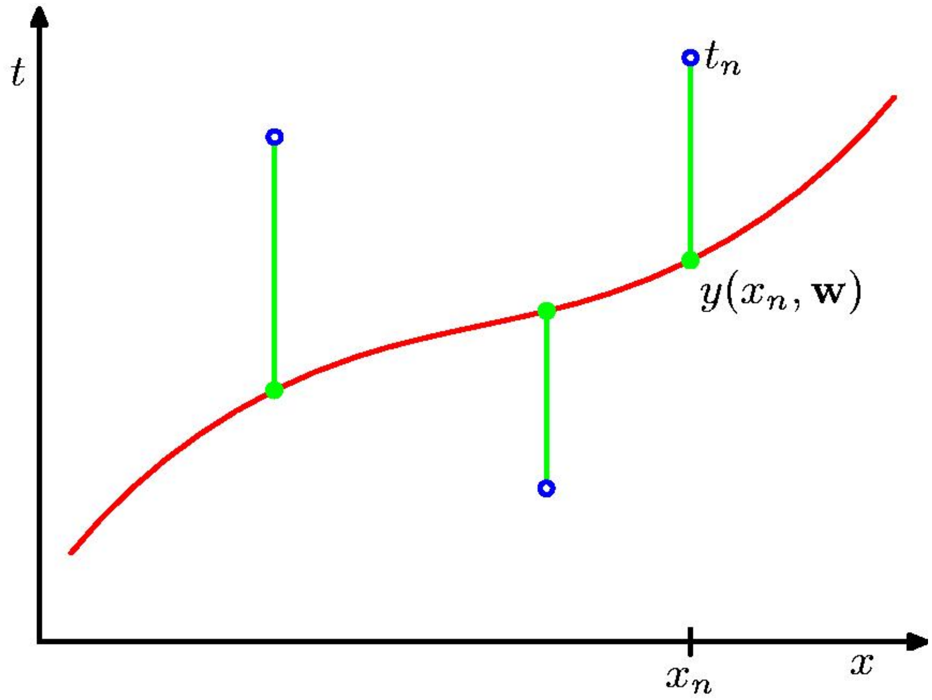
- What is learning?
  How do we learn?

Ex: Polynomial Curve Fitting



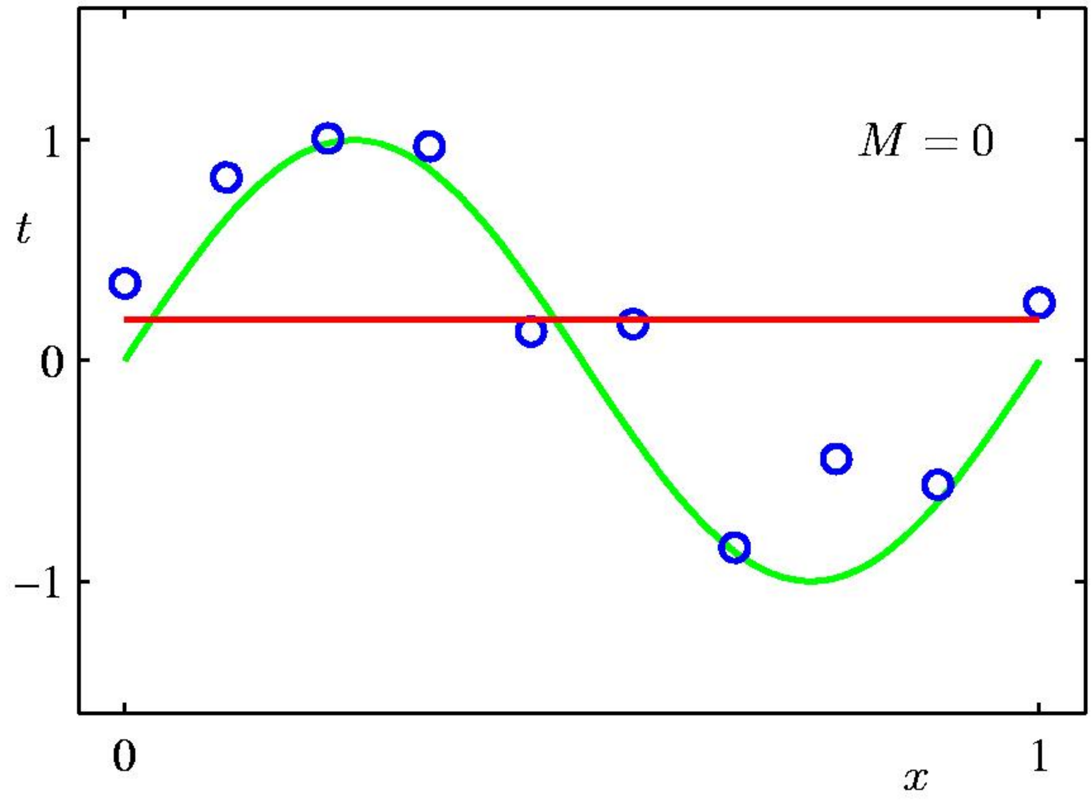
$$y(x, \mathbf{w}) = w_0 + w_1 x + w_2 x^2 + \ldots + w_M x^M = \sum_{j=0}^{M} w_j x^j$$

Sum-of-Squares Error Function

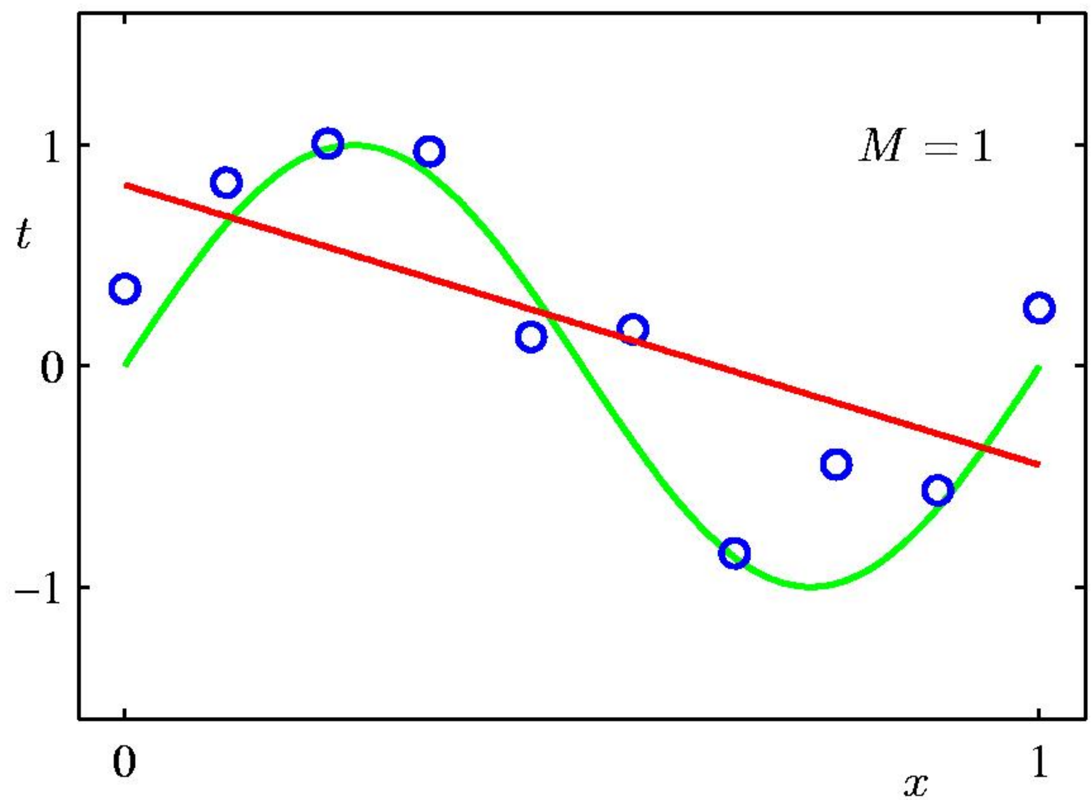


$$E(\mathbf{w}) = \frac{1}{2} \sum_{n=1}^{N} \{y(x_n, \mathbf{w}) - t_n\}^2$$

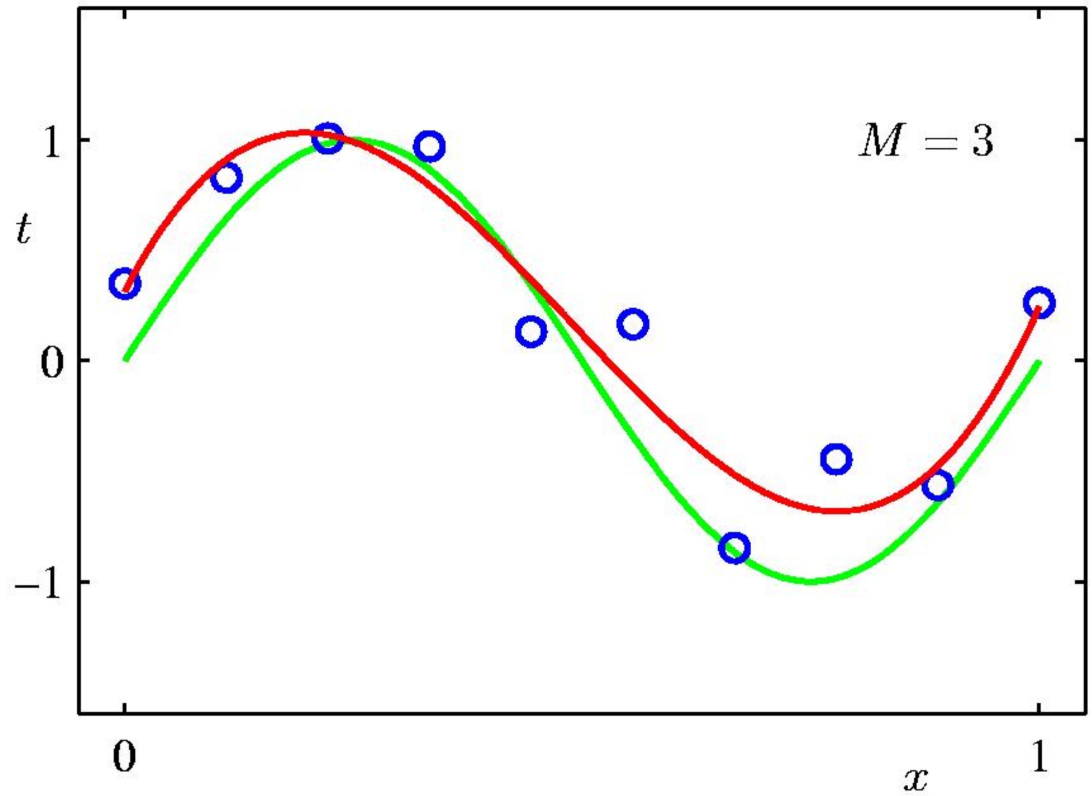
0^{th} Order Polynomial



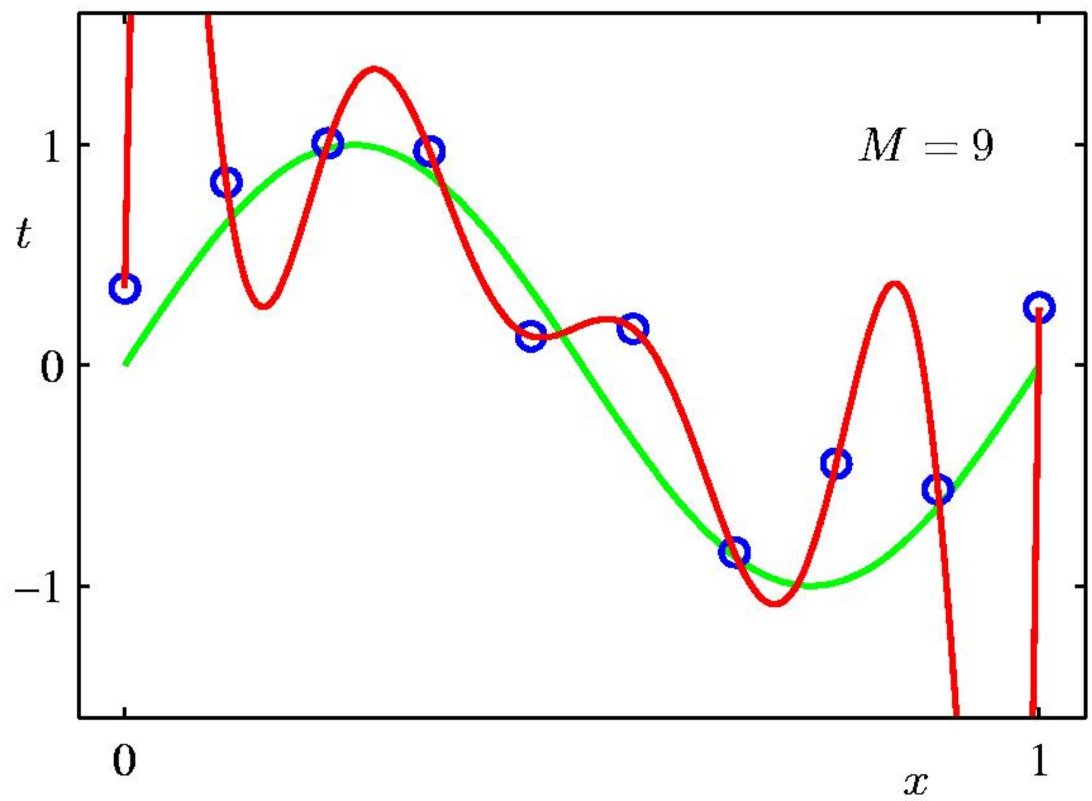
1st Order Polynomial



3rd Order Polynomial



9^{th} Order Polynomial



Which is better??

Output

- Errors
  Evaluation

Evaluation

- In general, define performance metric (e.g., Loss function), optimize

- In Supervised Learning:
  - Define error function
  - Learn a model on a **training set**
  - Evaluate model on a **test set**

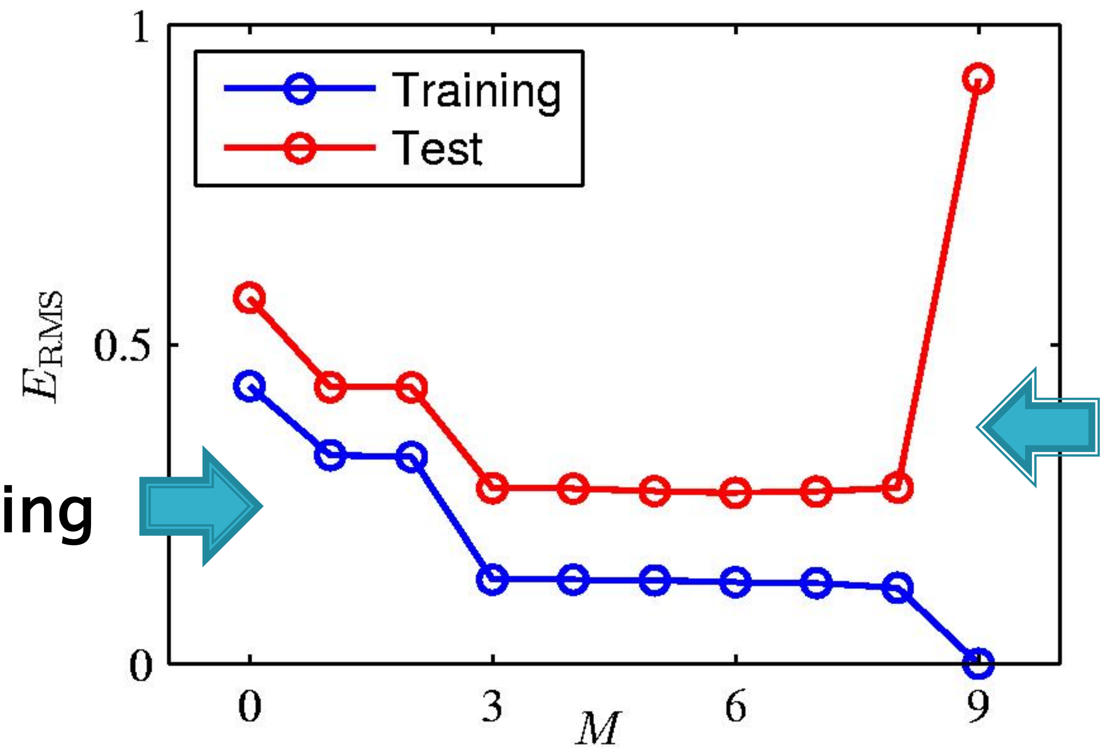
Ex. Spam Classification

- Two spam systems classify 100 emails
- System 1
  - 2 emails mistakenly labeled spam
  - 18 emails mistakenly labeled not-spam
- System 2
  - 10 emails mistakenly labeled spam
  - 0 emails mistakenly labeled not-spam
- Which is better?

Training Error for example

Polynomial Degree	Training Error	Test Error
0		
1		
2		
3		
9		

Under–fitting & Over–fitting



Under–fitting

Over–fitting

Root-Mean-Square (RMS) Error: $E_{\mathrm{RMS}} = \sqrt{2E(\mathbf{w}^{\star})/N}$

Approaches to Model Selection

- Regularization
- Information Theory
- Bayesian Approach

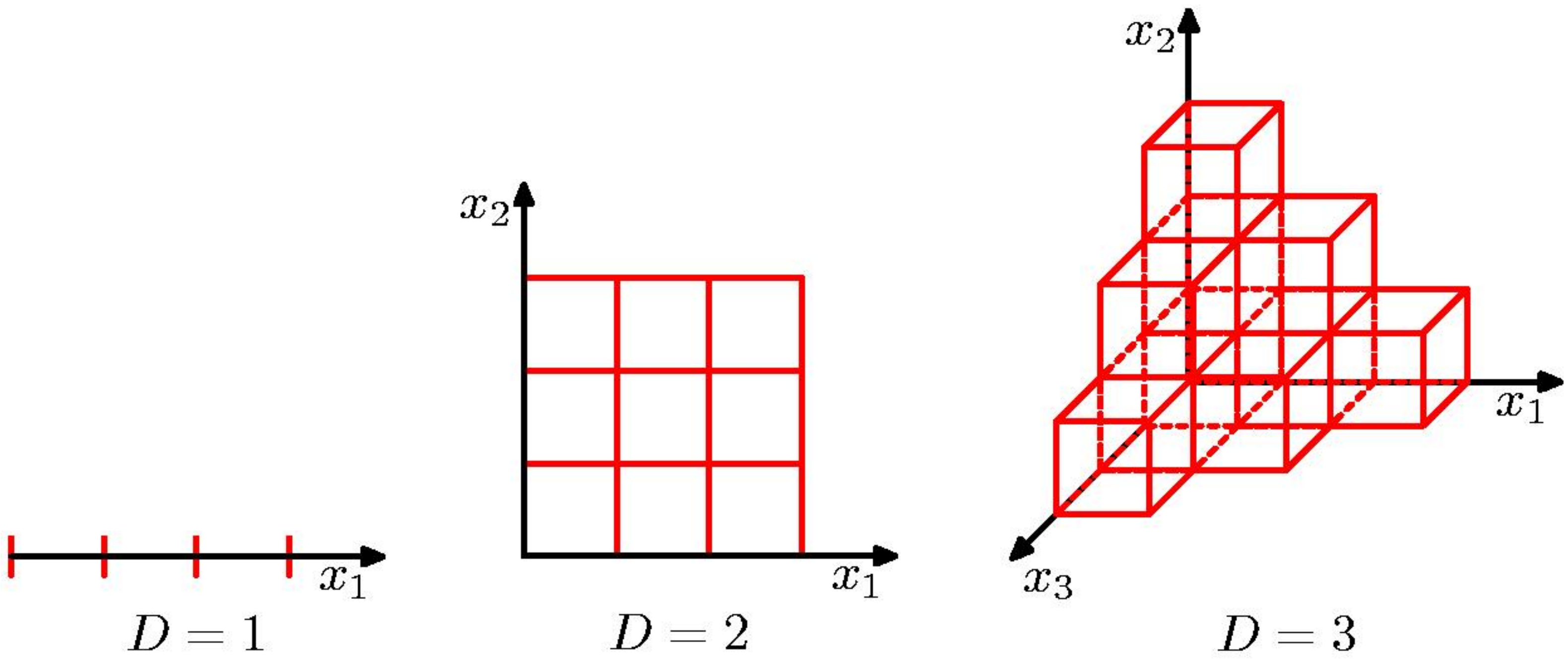
Examples of Learning Algorithms

- We'll spend most of the semester on this topic
- For each algorithm we'll study issues like
  - Algorithm's model of learning
  - How the model is trained
  - What loss functions are minimized

The Curse of Dimensionality

- Watch out!

Curse of Dimensionality



Curse of Dimensionality

Polynomial curve fitting, M = 3

$$y(\mathbf{x}, \mathbf{w}) = w_0 + \sum_{i=1}^{D} w_i x_i + \sum_{i=1}^{D} \sum_{j=1}^{D} w_{ij} x_i x_j + \sum_{i=1}^{D} \sum_{j=1}^{D} \sum_{k=1}^{D} w_{ijk} x_i x_j x_k$$

Places where intuitions fail…

- Almost all of the volume of a sphere is concentrated in its shell for high dimensional spheres. (In the book.)

- Almost all of the mass of a high dimensional Gaussian distribution is concentrated in a thin shell. (In the book.)

- Spheres in high dimensions don't even look round! They look like porcupines! (See http://mark.reid. name/iem/warning-highdimensions.html.)

- The ratio of the distance from the centre of the hypercube to one of the corners, divided by the perpendicular distance to one of the sides, is sqrt(D), which therefore goes to infinity as D goes to infinity. From these results we see that, in a space of high dimensionality, most of the volume of a cube is concentrated in the large number of corners, which themselves become very long spikes.

Quote…

“High dimensions are weird”

Hal Daume III

Survey Results

Summary: Key Ideas

- Be able to recognize ML Settings
- Training Set vs. Test Set
- Concepts of Under-fitting/Over-fitting
- Curse of Dimensionality

Next Time….

- Reading: Bishop 1.2, Bishop Appendix C
- Additional supplementary pointers on the website.