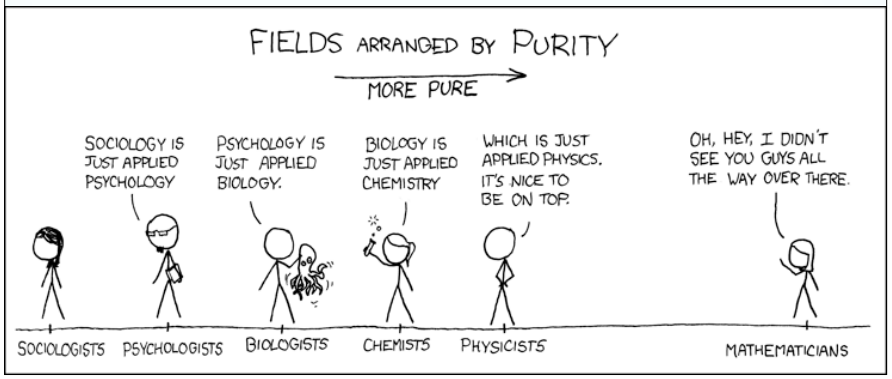
Welcome to the Wonderful World of Machine Learning

Mark Dredze
Machine Learning (CS 600.475)



Today's Topics

- Course Overview
  - Goals
  - Policies
- The Field of Machine Learning
  - What is it?
  - History
- Linear regression

Course Goals

- Learn the fundamentals of machine learning
- Learn to implement machine learning applications
- Learn how to apply machine learning to different settings

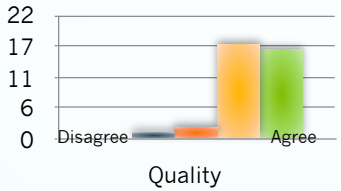
Course Policies

- Website: cs475.org
  - Bulletin Board
- Requirements
  - Programming: Python
  - Math
- Midterm
- Homeworks
  - Goals
  - Late Policy
- Final project
- Readings
- Grading
- Cheating

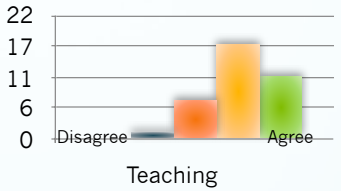
Todo List

- Get password for cs475.org account
- Sign up for an account on the message board
- Get a copy of the textbook
- Read about the course, including policies on the About section of the website
- Class registration
- Waitlist students: complete new poll

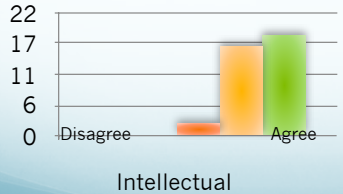
Course Ratings (Spring 2014)



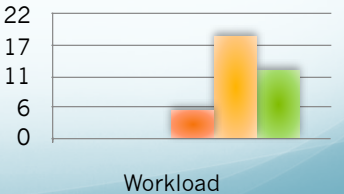
Quality



Teaching

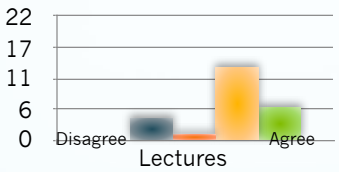


Intellectual

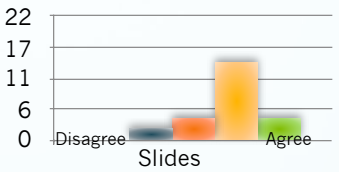


Workload

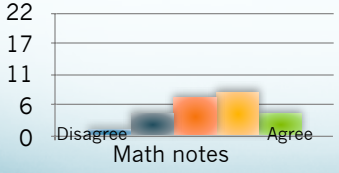
What Helped Learning?



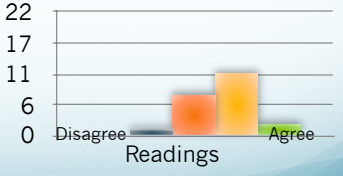
Lectures



Slides



Math notes



Readings

Machine Learning Foundations

Definition

- Machine learning allows computers to observe input and produce a desired output, either by example or through identifying latent patterns in the input.
- Data
  - What type of input?
  - What type of output?
- Patterns = algorithm
  - Intuition (empirical)
  - Objective (theoretical)

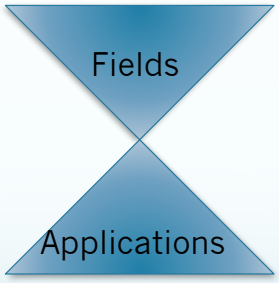
Different Definition

Fitting a function to data

- Fitting: Optimization, what parameters can we change?
- Function: Model, loss function
- Data: Data/model assumptions? How we use data?
- ML Algorithms: minimize a function on some data

A Diverse Topic

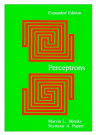
AI, Statistics, Linear algebra, Information Theory, Probability, Decision Theory



Vision, Robotics, NLP, Computational Biology, Bioinformatics, Speech, IR, Systems

Frank Rosenblatt (Cornell)
Linear classifier for character recognition
Based on neurons in the brain
By studying the Perceptron, "the fundamental laws of organization which are common to all information handling systems, machines and men included, may eventually be understood."

1959 1960 1969



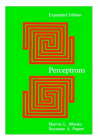
Marvin Minsky, Seymour Papert (MIT)
"Perceptrons: An Introduction to Computational Geometry"
Criticism of Perceptron demonstrating limitations
Research in Perceptrons stops for almost 20 years

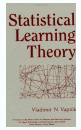
Arthur Samuel (IBM)
Checkers playing program
Rote learning- simulate many games to learn strategy
Early reinforcement learning

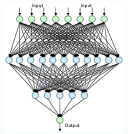
First workshop on Machine Learning
Second in 1983
Eventually became ICML in 1993

ICML

1959 1960 1969 1974 1980 1982



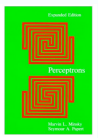


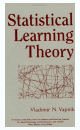
Multi-layer neural networks
1986· Backprop algorithm

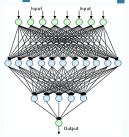
Vapnik and Chervonenkis
Statistical learning theory
Early idea behind SVMs
1990· kernels
1992· modern SVM

ICML

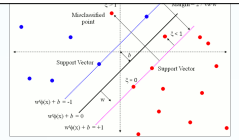
1959 1960 1969 1974 1980 1982 1985



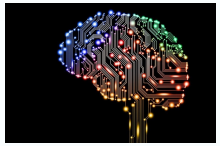


Rise of machine learning as a field
1985· AIStats
1985· UAI
1987·NIPS (90 papers, 2014: 1678 submissions, 414 papers)
1988· COLT



Rise of
the SVMs

3rd rise of
neural
networks
(now deep
learning)



1992 1993 2002 2007 2010 2016

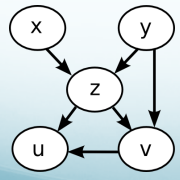
Second
death of
neural
networks

Rise of
the
graphical
models

Deep
learning
takes over

ML hits record size

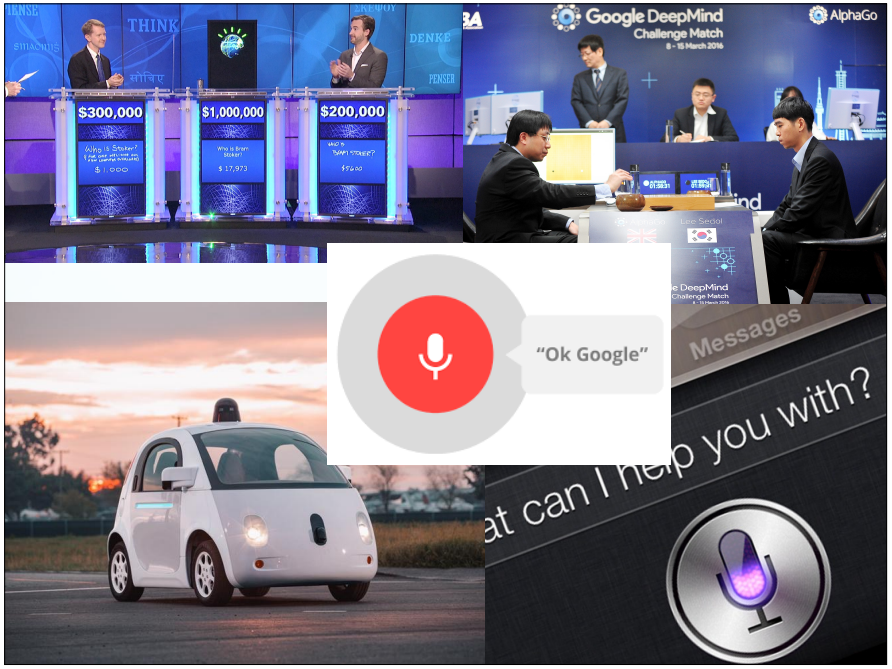
· 3000 people at NIPS 2015
· ICML 2016 doubles in size



Years are
approximate

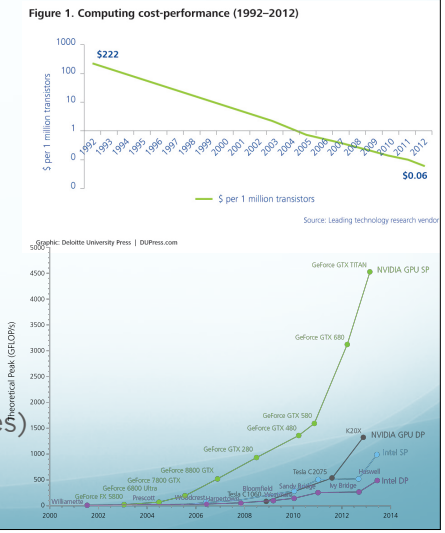
Machine Learning Today

- ~~Machine Learning~~
- Artificial intelligence
  - ML split from AI ~20 years ago, coming back
- Why?



Why success?

- (Somewhat) Better algorithms
  - Deep learning
- Better computers
- More data
- The rise of data science
  - More companies hire ML, more success (cycles)



ALLEN INSTITUTE *for* ARTIFICIAL INTELLIGENCE

Google DeepMind

OpenAI

Stanford | One Hundred Year Study on Artificial Intelligence (AI100)

Linear Regression
Our First Algorithm

If she loves you more each and every day,
by linear regression she hated you before you met.

Why Start Here?

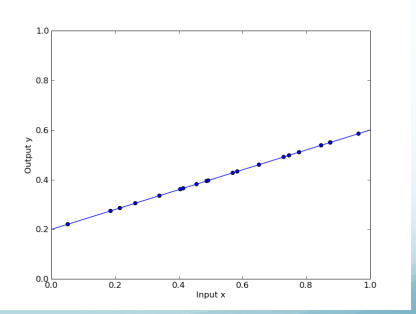
- Linear regression is well known
  - Not strictly a machine learning algorithm
- Learn about fundamentals of machine learning using a simple example

Example

- I have a large number of undergraduate applications for Johns Hopkins. I want to accept students who I think will get the highest GPA (0-4.0).
- Goal: Predict an applicant's GPA
- Data: Previous applications and resulting GPAs
- How do I do this?

Data Model

- Assume dependent variable (y) can be modeled by a *linear function* of the input variables (x)
- $y = \mathbf{w}\mathbf{x} + b$
- 2 dimensions
  - Compute w and b from two points
- Solution?
  - Given y and x, solve for w



Regression

- Data $\{(\mathbf{x}_i, y_i)\}_{i=1}^{N} \quad \mathbf{x}_i \in \Re^M \quad y_i \in \Re$ ← Continuous Values
- Learn: a mapping from x to real valued y
  - f(x) = y
- Examples
  - GPAs
  - Stock price
  - Miles per gallon
  - Age of author

Try it at Home!

- Linear regression demo
- http://mste.illinois.edu/users/exner/java.f/leastsquares/
- http://onlinestatbook.com/2/regression/linear_fit_demo.html
- https://www.geogebra.org/m/FUe3HfRf

Recall Definition

Fitting a function to data

- Fitting: Solve for w given y and x
- Function: linear function
- Data: assume dependent variable linear combination of independent variables
- Minimize a function
  - What function are we minimizing?

What is the goal?

- You probably know linear regression from statistics
  - "In statistics, linear regression is an approach for modeling the relationship between a scalar dependent variable y and one or more explanatory variables (or independent variables) denoted X." (Wikipedia)
- Our goal: predict correctly the next example
  - Minimize: reduce prediction error
    - More to say about this later

Loss Functions

- Machine learning algorithms minimize loss functions
  - Or some substitute for a loss function
  - The best solution minimizes the loss function*
- Definition
  - A function that maps between (true label, prediction) -> non-negative number
  - 0 = perfect prediction

* Maybe

Loss: What We Minimize

- Loss measures the badness of our prediction
- What's a good loss function?
  - It depends on task and goals
- Regression loss function?
  - Proposal: How far are you from the correct answer

Sum of Squares Loss

$$f(x) - y$$

$$(f(x) - y)^2$$

$$\sum_{i=1}^{n} (f(x_i) - y_i)^2$$

GPA Example
$(3.5\text{-}3.0)^2 = .25$

Over all answers

Correct answer

$$\sum_{i=1}^{n} (w \cdot x_i - y_i)^2$$

Predicted answer

Recall Definition

Fitting a function data

- Fitting: Solve for w given y and x
- **Function: linear function**
- Data: assume dependent variable linear combination of independent variables
- Minimize a function
  - What function are we minimizing?

Hypothesis Class

- Learning algorithm selects hypothesis from hypothesis class
- Hypothesis class
  - A set of possible hypotheses (functions) that can be used to label the data
  - Can be finite or infinite
- Each learning algorithm has a hypothesis class
  - Fitting selects the best hypothesis using observed data

What is best hypothesis?

Choosing Hypothesis Class

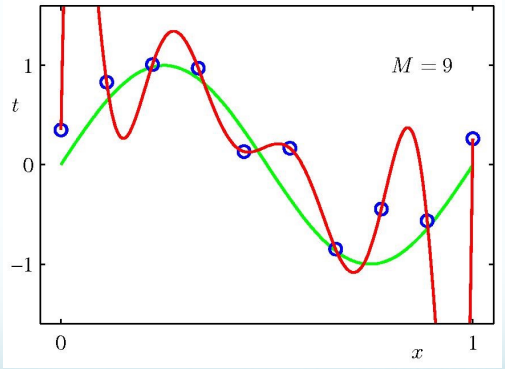
- What sort of hypothesis class do we want?
- The hypothesis class should contain the optimal hypothesis
- The hypothesis class for which our algorithm will find the best performing hypothesis

What is the difference?

Hypothesis Tradeoff

- Rich hypothesis class
  - Over-fitting
- Easier to search hypothesis class
  - Under-fitting
- Realization: we are unlikely to ever find the hypothesis that exactly explains the data
- Simplifying assumptions help find a reasonable hypothesis
- Select hypothesis class based on knowledge of data

Under/Over Fitting



Hypothesis Class

- What is the hypothesis class of linear regression?
- Linear functions
  - All possible linear functions encoded by parameters w
  - Hypothesis chosen by setting parameter values w
- How large is the hypothesis class?
  - Infinite

What is Learning? Another View

- Select a hypothesis from the hypothesis class
  - Model parameters correspond to hypotheses
  - Learn parameters of model based on data
- How do we write learning algorithms?
  - Theory: Objective driven
    - Write an objective that you want to minimize
    - Develop a procedure to minimize the objective
      - This is called the learning algorithm
  - Empirical: intuition driven
    - Many algorithms based on motivation and heuristics
    - Post hoc analysis of objective

Recall Definition

Fitting a function to data

- Fitting: Solve for w given y and x
- Function: linear function
- **Data: assume dependent variable linear combination of independent variables**
- Minimize a function
  - What function are we minimizing?

Learning Settings

- What information is available for learning?
  - What does the data look like?
  - How is it annotated?
- What output is desired?
  - What should the algorithm produce?
  - How will it be used?

Supervised Learning

- Learning with a teacher
  - Explicit feedback in the form of labeled examples
  - Goal: make prediction
  - Pros: Good performance
  - Cons: Labeled data is difficult to find
- Examples
  - Regression!
  - Classification
    - Sort documents by topic
  - 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

Semi-Supervised Learning

- Labeled examples + 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
  - Piloting an airplane
  - Go

How Do We Represent Data?

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

High Dimensional Vectors

- A learning example is a vector of length M x
- Examples a drawn from an underlying distribution

$$x_i \in \Re^M$$

- Each dimension represents a feature
  - Feature functions

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

- A collection of N examples

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

What Does Data Look Like?

Representations

$$x$$

- Designing feature functions is critical
  - Well designed representations greatly effect performance
- How should we 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
- More on this in last lecture

Recall

- Our goal: predict correctly the next example
  - Minimize: reduce prediction error

Goal of Learning

- "Reduce prediction error"
  - On what?
- True error

$$error_D(h) = P_{x \in D}[\ell(h(x_i), y_i) \neq 0]$$

  - We need infinite data to measure this!

Goal of Learning

- If we can't measure true error, how do we judge learning success?
- Should an algorithm maximize performance on observed data?
- Proposal: measure error on the given data
  - Call this the "training data"
  - Is this a good idea?

Measuring Error

- Very bad idea
- Recall: machine learning cares about prediction (the future)
  - How well will the system do once deployed?
- Memorizing the training data is easy
  - Most hypothesis classes are rich enough to exactly learn the training data

Difficult to Understand

- People routinely make this mistake!
- "The team from the eastern most state has always won the Superbowl when their quarterback is taller and the temperature is above 60°"
- True stories from my experiences
  - Project accepted to predict hot real estate markets
    - Promising because very high accuracy
    - Problem: measured error on training data
  - Paper submission to major conference claimed to have solved problem with 100% accuracy
    - Trained on the test data
  - Paper submission showed high accuracy on classification task
    - Data *written* by researchers with task in mind

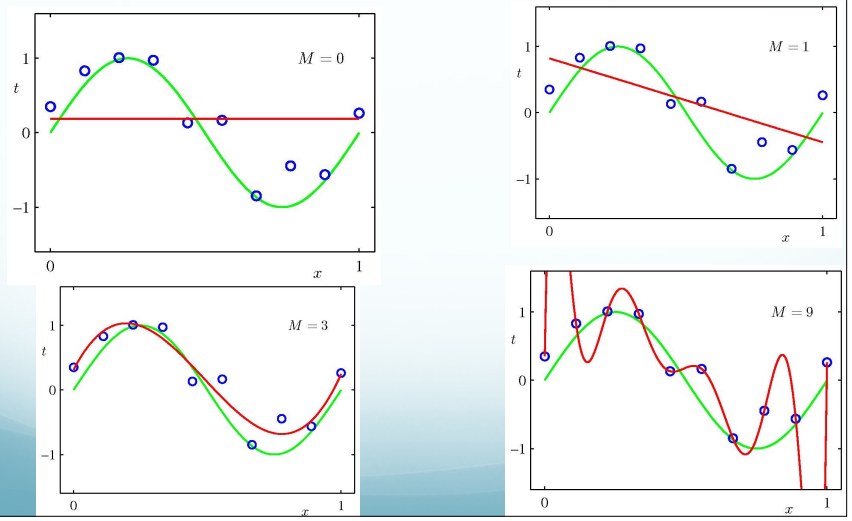
Generalization

- Generalization
  - The ability of an algorithm to generalize knowledge learned from observed data to new data
- Simple example: memory based classifier
  - Binary classification
  - Train: remember each example
  - Test: if we have seen an example before, report label
    - Otherwise, guess randomly
- Train error: 0% Test error: 50%
  - This is called over-fitting

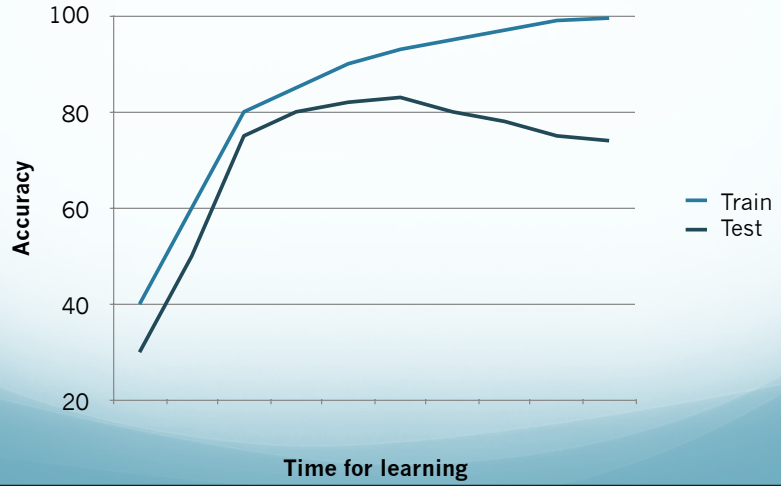
Bias vs. Variance

- How do we achieve good generalization?
- Tradeoff bias and variance
  - Bias- favor certain predictions
  - Variance- diversity of predictions
- Under-fit observed data
  - Favors bias- large changes to input have same output
- Over-fit observed data
  - Favors variance- small changes to input can dramatically change output
- Tradeoff key to generalization to new data

Under/Over Fitting



Typical Learning Curves



Measuring Error: The Right Way

- Collect two sets of data
  - Train data- use for training algorithm
  - Test data- only use for evaluation
    - Only good if you've never seen it before, not if you continuously tune on it
- How do we balance bias/variance?
  - Tune parameters on development/validation data

Two Common Methods

- Train/dev/test
  - Use held out sets for evaluation
  - Good when you have lots of data
- Cross validation
  - Create many train/test splits
  - Randomly sample train and test data splits
  - Divide data into folds, use each fold for testing once
  - Reasonable when you don't ave enough data

What Causes Error?

- Noise error
  - An example has an incorrect or inconsistent label
  - Our data representation fails to encode necessary information
- Model error
  - Hypothesis class is deficient
- Parameter estimation error
  - The model parameters are wrong
- Search error
  - We made a mistake in scoring a prediction
  - Common in tasks with complex output

Recall Definition

Fitting a function to data

- **Fitting: Solve for w given y and x**
- Function: linear function
- Data: assume dependent variable linear combination of independent variables
- Minimize a function
  - What function are we minimizing?

Linear Regression

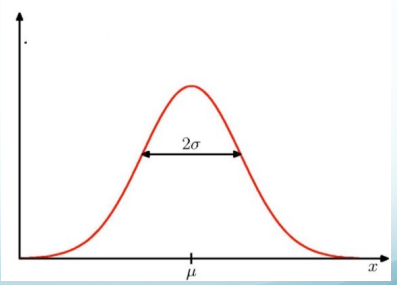
- We assumed output (y) linear combination of inputs
- This is wrong
  - Rarely is data actually linear
- But realistic assumption may be too complex
- A reasonable middle ground?

Noise from a Gaussian Distribution

$$N(x \mid \mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{-\frac{1}{2\sigma^2}(x-\mu)^2\right\}$$

$$E[x] = \mu$$

$$\mathrm{var}[x] = \sigma^2$$



Noise

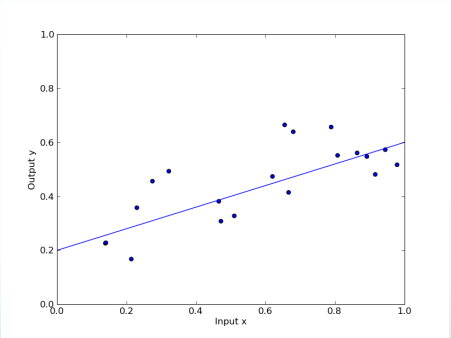
- Assume output permuted by Gaussian noise

$$y = h_w(x) + \varepsilon$$

$$\varepsilon \sim N(\mu, \sigma^2), \quad \mu = 0, \quad \sigma^2 = 1$$

- The data isn't really generated in this way
  - Assume that it is for sake of modeling

Linear Regression with Noise



Probability of Output

$$p(y \mid x, w, \sigma^2) = N(y, \sigma^2)$$
$$= N(h_w(x), \sigma^2)$$

Least Squares Regression

- Fitting: Solve for w given x and y
- Function: Linear function + Gaussian noise
  - Loss function: squared error
  - Assumes output (mostly) a linear combination of input
- Data: fit a model to training data, evaluate on held out data
- Minimize a function
  - What function are we minimizing?

Which Function are we Minimizing?

- Which is the best hypothesis?
  - Which setting of the parameters w is best?
- Select the hypothesis that *best explains* the observed data
- Select the hypothesis that minimizes the error

Explaining the Data

- What does it mean to explain the data?
  - Maximize the likelihood of the data
  - Likelihood = probability of observing data
- Writing likelihood
  - Assume data generated from our linear regression model

Maximum Likelihood For Gaussians

Maximum Likelihood for Regression

Sources of Error

- Noise error
  - An example has an incorrect or inconsistent label
  - Our data representation fails to encode necessary information
- Model error
  - Hypothesis class is deficient
- Parameter estimation error
  - The model parameters are wrong
- Search error
  - We made a mistake in scoring a prediction
  - Common in tasks with complex output

Bias?

- Gaussians: maximum likelihood estimate is biased
  - This is ok if we have infinite data
  - We never have infinite data!
- Over-fitting: avoid it by favoring certain solutions
- Regularization
  - Add term to objective to favor different considerations
  - What should we favor?
    - Occam's Razor: simpler is better
    - Favor small weights
      - Our parameters should be small

Regularized Least Squares

Regularized Least Squares

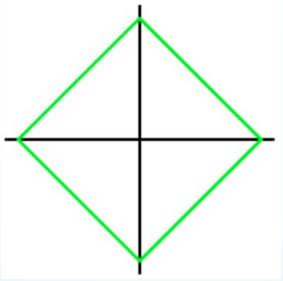
- Tradeoff: low error and small weights

$$E_D(w) = \frac{1}{2}\sum_{i=1}^{n}(y - w \cdot x)^2$$

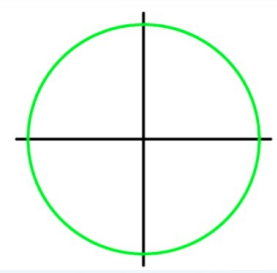
$$E_W(w) = \frac{1}{2}w^T w$$

$$E_D(w) + \lambda E_W(w) = \frac{1}{2}\sum_{i=1}^{n}(y - w \cdot x)^2 + \lambda\frac{1}{2}w^T w$$

Regularization Behavior



q=1 L1 (Lasso)



q=2 L2 (quadratic)

Bias vs. Variance

- Expected squared loss can be written as

$$\mathrm{E}[L] = \int \{f(x) - h(x)\}^2 p(x)dx + \int \{h(x) - y\}^2 p(x,y)dx\,dt$$

  - f(x)- prediction function
  - h(x)- true regression function
  - y- provided label (noisy)

  - First term- minimize loss
  - Second term- error from noise

Bias vs. Variance

- Imagine we can sample many datasets D from the underlying distribution
- Integrate the first term (which represented accuracy of the model)

$$(f(x,D) - h(x))^2$$

  - Depends on a particular sample of data D
- What is the expectation of this term over many samples of D?

Bias vs. Variance

$$\mathrm{E}_D[(f(x,D) - h(x))^2] =$$
$$(\mathrm{E}_D[f(x,D)] - h(x))^2 + \mathrm{E}_D[(f(x,D) - \mathrm{E}_D[f(x,D)])^2]$$

Bias Variance

- For learning we want to minimize this function
- The result is a tradeoff between bias and variance

Parameter Tradeoff

- The regularization parameter controls bias vs. variance
- Higher λ = more regularization
  - Favors bias
- Lower λ = less regularization
  - Favors variance

Problems

- Maximum likelihood under-estimates variance and over-fits
  - Try to fix using regularization
- How do we decide model complexity?
  - Parameter tuning on held out data
- Is there a better way?
  - Bayesian methods
  - more on this later

Summary: Machine Learning Fundamentals

Fitting a function to data

- Fitting: Optimization, what parameters can we change?
- Function: Model, loss function
- Data: Data/model assumptions? How we use data?
- ML Algorithms: minimize a function on some data

Summary: Machine Learning Fundamentals

- Data representation $\{(x_i, y_i)\}_{i=1}^{N} \quad x_i \in \Re^M \quad y_i \in \Re$
- Loss functions
- Hypothesis class and tradeoffs
- Generalization and bias/variance tradeoff
- Learning settings: Supervised and unsupervised
- Regularization
- Sources of error

Next Time: On to Classification!