ECE 5984: Introduction to Machine Learning

Topics:

- Ensemble Methods: Bagging, Boosting

Readings: Murphy 16.4; Hastie 16

Dhruv Batra

Virginia Tech

Administrativia

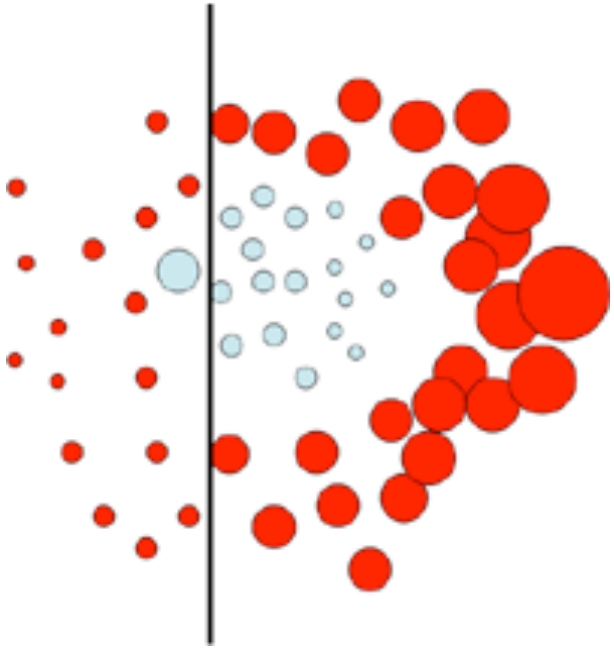
- HW3
  - Due: April 14, 11:55pm
  - You will implement primal & dual SVMs
  - Kaggle competition: Higgs Boson Signal vs Background classification
  - https://inclass.kaggle.com/c/2015-Spring-vt-ece-machine-learning-hw3
  - https://www.kaggle.com/c/higgs-boson

Administrativia

- Project Mid-Sem Spotlight Presentations
  - 9 remaining
  - Resume in class on April 20th
  - Format
    - 5 slides (recommended)
    - 4 minute time (STRICT) + 1-2 min Q&A
  - Content
    - Tell the class what you're working on
    - Any results yet?
    - Problems faced?
  - Upload slides on Scholar

New Topic: Ensemble Methods

Bagging



Boosting

Synonyms

- Ensemble Methods
- Learning Mixture of Experts/Committees
- Boosting types
  - AdaBoost
  - L2Boost
  - LogitBoost
  - <Your-Favorite-keyword>Boost

A quick look back

- So far you have learnt

- Regression
  - Least Squares
  - Robust Least Squares

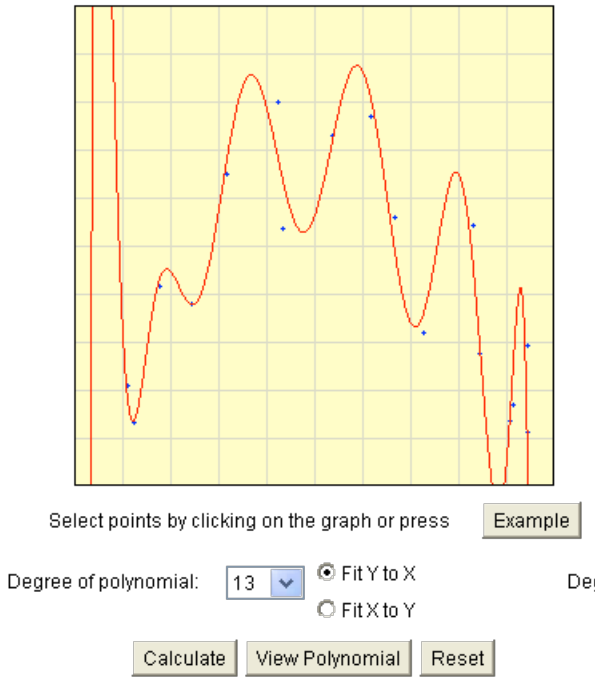
- Classification
  - Linear
    - Naïve Bayes
    - Logistic Regression
    - SVMs
  - Non-linear
    - Decision Trees
    - Neural Networks
    - K-NNs

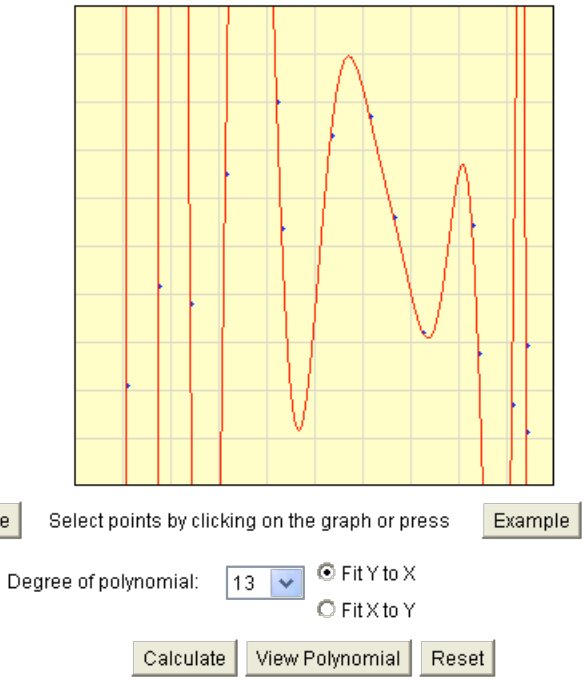
Recall Bias-Variance Tradeoff

- Demo
  - http://www.princeton.edu/~rkatzwer/PolynomialRegression/

Bias-Variance Tradeoff

- Choice of hypothesis class introduces learning bias
  - More complex class → less bias
  - More complex class → more variance





Fighting the bias-variance tradeoff

- **Simple (a.k.a. weak) learners**
  - e.g., naïve Bayes, logistic regression, decision stumps (or shallow decision trees)
  - **Good:** Low variance, don't usually overfit
  - **Bad:** High bias, can't solve hard learning problems

- **Sophisticated learners**
  - Kernel SVMs, Deep Neural Nets, Deep Decision Trees
  - **Good:** Low bias, have the potential to learn with Big Data
  - **Bad:** High variance, difficult to generalize

- Can we make combine these properties
  - **In general, No!!**
  - **But often yes…**

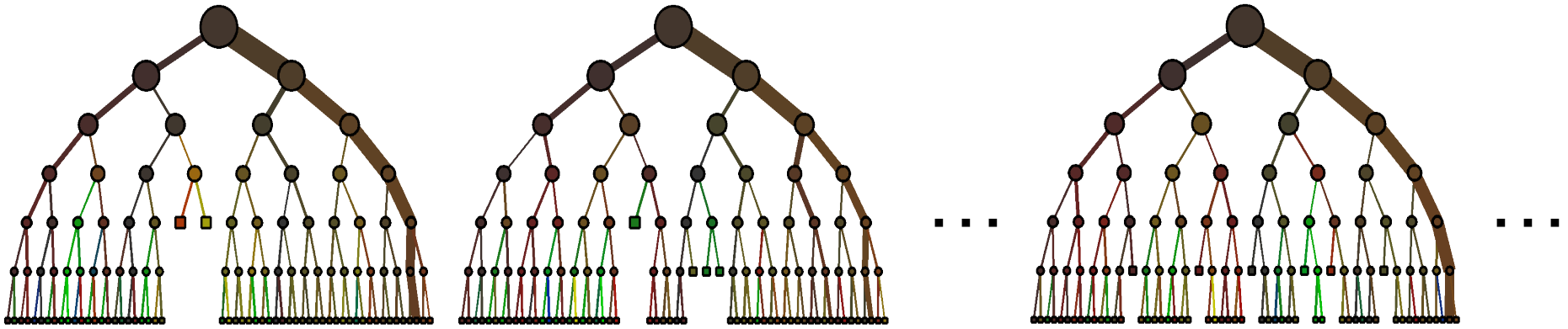
Voting (Ensemble Methods)

- Instead of learning a single classifier, learn **many classifiers**
- **Output class:** (Weighted) vote of each classifier
  - Classifiers that are most "sure" will vote with more conviction
- With sophisticated learners
  - Uncorrelated errors → expected error goes down
  - On average, do better than single classifier!
  - Bagging
- With weak learners
  - each one good at different parts of the input space
  - On average, do better than single classifier!
  - Boosting

Bagging

- Bagging = Bootstrap Averaging
  - On board
  - Bootstrap Demo
    - http://wise.cgu.edu/bootstrap/

Decision Forests



Learn many trees & Average Outputs

Will formally visit this in Bagging lecture

Boosting [Schapire, 1989]

- Pick a class of weak learners $\mathcal{H} = \{h \mid h : X \to Y\}$
- You have a black box that picks best weak learning
  - unweighted sum $h^* = \underset{h \in \mathcal{H}}{\operatorname{argmin}} \sum_i L\left(y_i, h(x_i)\right)$
  - weighted sum $h^* = \underset{h \in \mathcal{H}}{\operatorname{argmin}} \sum_i w_i L\left(y_i, h(x_i)\right)$
- On each iteration *t* $f_{t-1}(x_i) = \sum_{t'=1}^{t} \alpha_{t'} h_{t'}(x_i)$
  - Compute error for each point
  - Update weight of each training example based on it's error.
  - Learn a hypothesis – h_t
  - A strength for this hypothesis – α_t
- Final classifier: $f_t(x) = f_{t-1} + \alpha_t h_t(x)$

Boosting

- Demo
  - Matlab demo by Antonio Torralba
  - http://people.csail.mit.edu/torralba/shortCourseRLOC/boosting/boosting.html

Types of Boosting

Loss Name	Loss Formula	Boosting Name
Regression: Squared Loss	$(y - f(x))^2$	L2Boosting
Regression: Absolute Loss	$\|y - f(x)\|$	Gradient Boosting
Classification: Exponential Loss	$e^{-yf(x)}$	AdaBoost
Classification: Log/Logistic Loss	$\log\left(1 + e^{-yf(x)}\right)$	LogitBoost

L2 Boosting

Loss Name	Loss Formula	Boosting Name
Regression: Squared Loss	$(y - f(x))^2$	L2Boosting

- Algorithm
  - On Board

Adaboost

Loss Name	Loss Formula	Boosting Name
Classification: Exponential Loss	$e^{-yf(x)}$	AdaBoost

- Algorithm
  - You will derive in HW4!
- Guaranteed to achieve zero training error
  - With infinite rounds of boosting (assuming no label noise)
  - Need to do early stopping

What you should know

- Voting/Ensemble methods
- Bagging
  - How to sample
  - Under what conditions is error reduced
- Boosting
  - General outline
  - L2Boosting derivation
  - Adaboost derivation

Learning Theory

Probably Approximately Correct (PAC) Learning

What does it formally mean to learn?

Learning Theory

- We have explored **many** ways of learning from data

- But…
  - How good is our classifier, really?
  - How much data do I need to make it “good enough”?

A simple setting…

- Classification
  - N data points
  - **Finite** space H of possible hypothesis
    - e.g. dec. trees of depth d

- A learner finds a hypothesis *h* that is **consistent** with training data
  - Gets zero error in training – $error_{train}(h) = 0$

- What is the probability that *h* has more than ε true error?
  - $error_{true}(h) \geq \varepsilon$

Generalization error in finite hypothesis spaces [Haussler '88]

- ***Theorem***:
  - Hypothesis space *H* finite
  - dataset *D* with *N* i.i.d. samples
  - $0 < \varepsilon < 1$

For any learned hypothesis *h* that is consistent on the training data:

$$P(\text{error}_{true}(h) > \epsilon) \leq |H| e^{-N\epsilon}$$

Even if *h* makes zero errors in training data, may make errors in test

Using a PAC bound

$$P(\text{error}_{true}(h) > \epsilon) \leq |H|e^{-N\epsilon}$$

- Typically, 2 use cases:
  - 1: Pick ε and δ, give you *N*
  - 2: Pick N and δ, give you ε

Haussler '88 bound

$$P(\text{error}_{true}(h) > \epsilon) \leq |H| e^{-N\epsilon}$$

- Strengths:
  - Holds for all (finite) H
  - Holds for all data distributions

- Weaknesses
  - Consistent classifier
  - Finite hypothesis space

Generalization bound for |H| hypothesis

- ***Theorem***:
  - Hypothesis space *H* finite
  - dataset *D* with *N* i.i.d. samples
  - $0 < \varepsilon < 1$

For any learned hypothesis *h*:

$$P\left(\text{error}_{true}(h) - \text{error}_{train}(h) > \epsilon\right) \leq |H| e^{-2N\epsilon^2}$$

PAC bound and Bias-Variance tradeoff

$$P\left(\text{error}_{true}(h) - \text{error}_{train}(h) > \epsilon\right) \leq |H| e^{-2N\epsilon^2}$$

or, after moving some terms around, with probability at least 1-δ:

$$\text{error}_{true}(h) \leq \text{error}_{train}(h) + \sqrt{\frac{\ln |H| + \ln \frac{1}{\delta}}{2N}}$$

Important: PAC bound holds for all *h*,

but doesn't guarantee that algorithm finds best *h*!!!