Ensemble Learning

CSCI 447/547 MACHINE LEARNING

Outline

- Introduction
- Ideas:
  - Use mistakes to train subsequent models
  - Use many learners
  - Use simple algorithms (for now)
  - Weight sample contribution to error
  - Weight the learners
  - Change weights
- Conclusion

Types of Learning

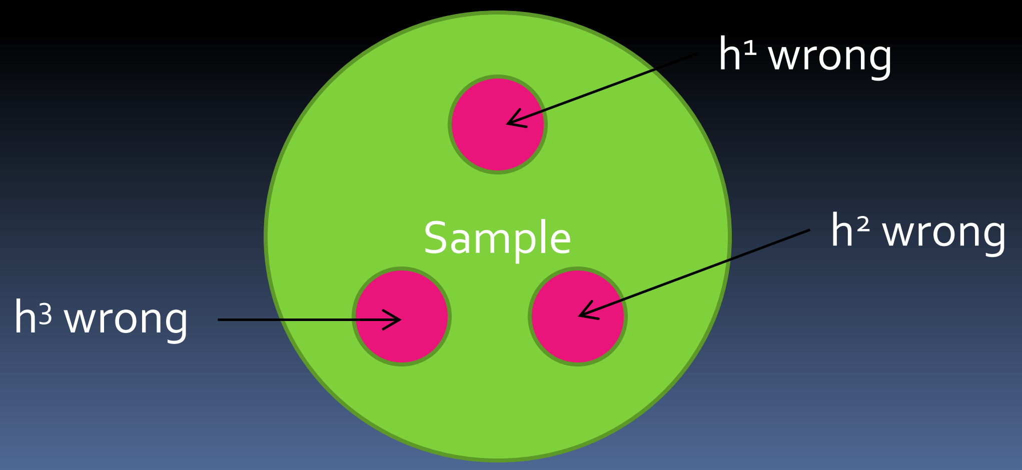
- Traditional
  - Regression
  - Nearest Neighbor
  - Decision Tree
- Biologically Inspired
  - Neural Nets
    - Local maxima, overfitting, oscillation
  - Genetic Algorithms
    - Naïve in attempt to mimic nature
- Theorists
  - Ensemble Learning
    - Can a crowd be smarter than the individuals in the crowd?

Binary Classification

- Classifier:
  - h [-1, +1]
  - Error rate will range from 0-1
    - Want an error rate of 0, it's bad if error rate is closer to 1
    - But what if error is close to, but a bit better than, chance?

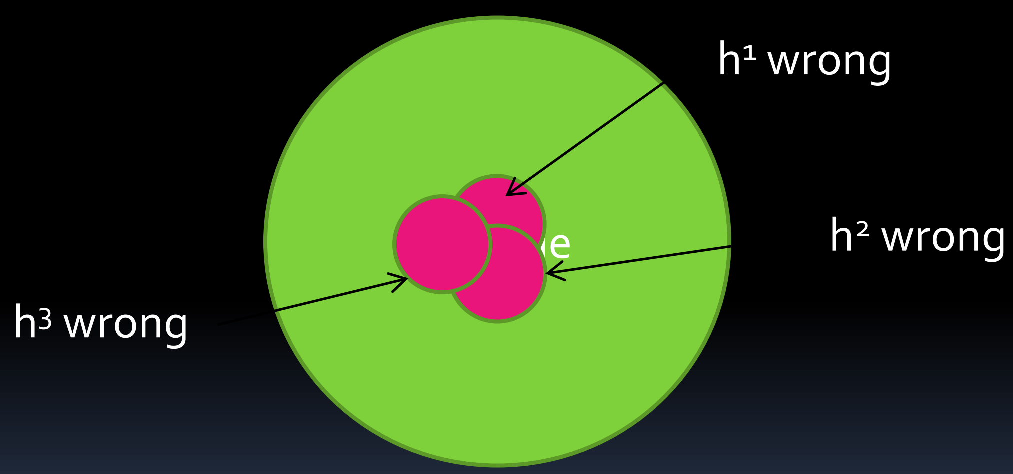
Classifier Errors

- Can we make a strong classifier out of weaker ones?
  - For example, make three classifiers and have them "vote"
    - $H(x) = \text{sign}(h^1(x) + h^2(x) + h^3(x))$



Classifier Errors

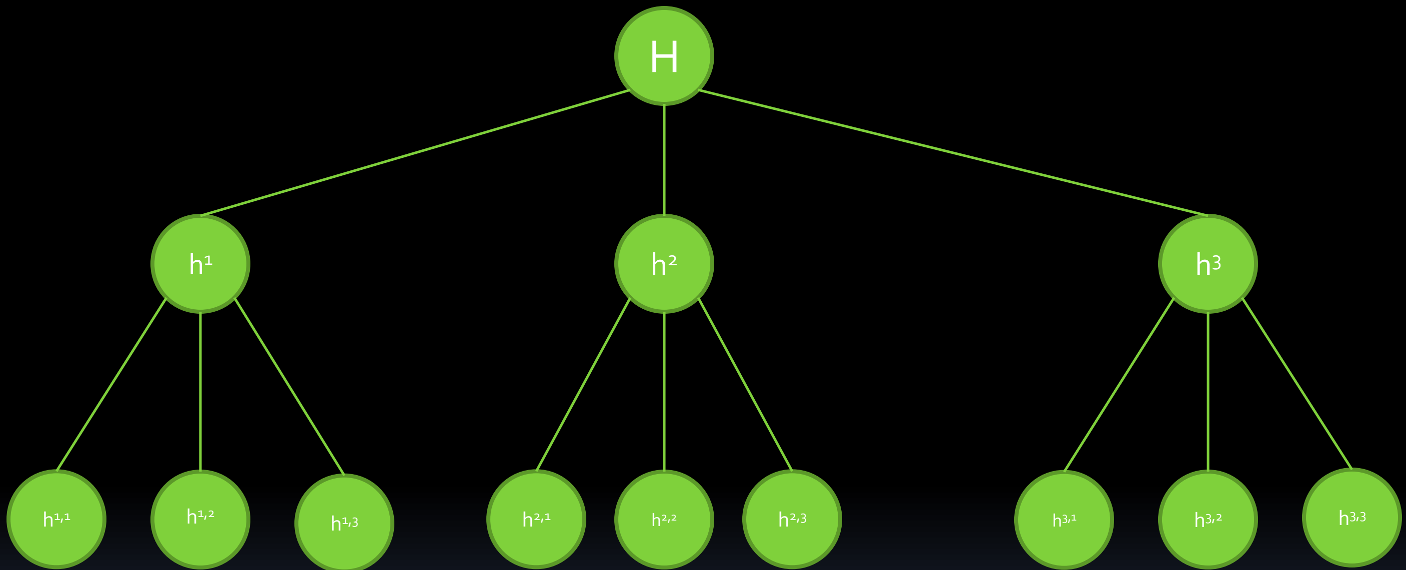
- But what if the results look more like this?



Idea 1

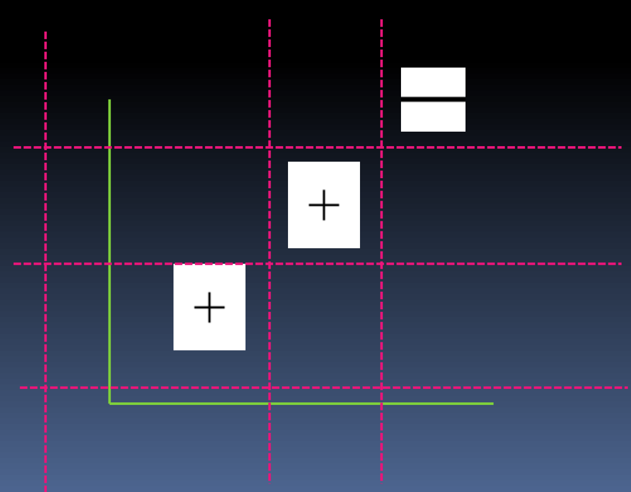
- Use our data to produce the best h^1 we can
- Then use the data with an exaggeration of h^1 errors to produce h^2
- Then use the data with an exaggeration of h^2 errors to produce h^3

Idea 2: Get Out the Vote



Idea 3: Simple Classifiers

- What kind of classifiers?
  - Coin flip?
  - Decision tree stumps
    - Single test tree

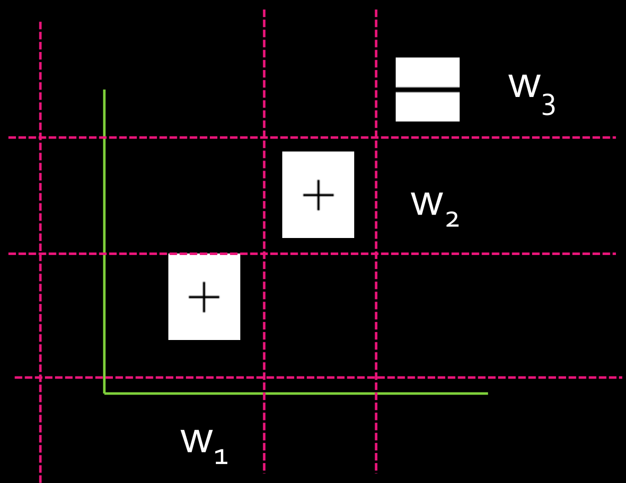


Only Stumps?!?

- No.
- Can use any kind of classifier you want
- Just easier to show derlooking at stumps

Idea 4: Samples Have Weights

$$Error = \sum_{wrong} \frac{1}{N}$$



- Assert each sample has a weight associated with it

$$w_i^1 = \frac{1}{N}$$

$$Error = \sum_{wrong} \frac{1}{N} = \sum_{wrong} w_i^1$$

- If we require:

$$\sum w_i = 1$$

  - this is a distribution

Idea 5: Wisdom of a (Weighted) Crowd (of Experts)

- $H(x) = \text{sign}(\alpha^1 h^1(x) + \alpha^2 h^2(x) + \alpha^3 h^3(x) + \ldots)$
- Let: $w_i^1 = \frac{1}{N}$
- Pick h^t that minimizes the error at time t
- Pick α^t
- Calculate W^{t+1}

Idea 6: Changing Weights

- Suppose $$w_i^{t+1} = \frac{w_i^t}{Z} e^{-\alpha^t h^t(x) y(x)}$$
- (Mathematical convenience)
- Z is a normalizing constant
- y(x) is just +1 or -1 depending on whether the output should be +1 or -1
  - If h(x) is correct then result is positive
  - If h(x) is wrong, will get a negative number

Changing Weights

- Want to find a way to minimize the error of the total expression
  - Minimum bound on the error for H(x) is:

$$\alpha^t = \frac{1}{2} ln \frac{1 - error^t}{error^t}$$

  - Yes, this means that the error can actually go up as you add terms to this ensemble function
  - All you know is the error rate is bounded by an exponential function

Changing Weights

$$w_i^{t+1} = \frac{w_i^t}{Z}\begin{cases} \sqrt{\frac{\epsilon^t}{1-\epsilon^t}}, & \text{Correct} \\ \sqrt{\frac{1-\epsilon^t}{\epsilon^t}}, & \text{Wrong} \end{cases}$$

$$Z = \sqrt{\frac{\epsilon^t}{1-\epsilon^t}} \sum_{correct} w_i^t + \sqrt{\frac{1-\epsilon^t}{\epsilon^t}} \sum_{wrong} w_i^t$$

$$\sum_{wrong} w_i^t = \epsilon^t$$

$$\sum_{correct} w_i^t = 1 - \epsilon^t$$

$$Z = 2\sqrt{\epsilon^t(1-\epsilon^t)}$$

$$w_i^{t+1} = \frac{w^t}{2}\frac{1}{1-\epsilon^t} \text{ if corrrect}$$

$$w_i^{t+1} = \frac{w^t}{2}\frac{1}{\epsilon^t} \text{ if wrong}$$

TGH #1

- Take all the weights from the previous test and where I got the right answer and add them up, and scale them so they sum to ½

$$\frac{1}{2}\frac{1}{1-\epsilon}(1-\epsilon) = \frac{1}{2}$$

$$\sum_{correct} w^{t+1} = \frac{1}{2}$$

$$\sum_{wrong} w^{t+1} = \frac{1}{2}$$

TGH #2

- How many tests to consider per decision tree stump?
  - Remember, each stump looks at one feature
  - Tests = Number of samples minus 1
  - But… don't need to consider tests that split adjacent samples of same class

Overfitting

- Almost ever type of classifier can overfit
- ***But***: Boosting does not appear to overfit
  - Experimental result, not mathematically proven
    - Each classifier divides space into chunks
    - Outliers can cause overfitting
  - Thought to be because boundaries around outliers end up so small that they don't influence actual class boundaries

Conclusion

- Magic
- Always want to use it
- Works with any kind of classifier

Summary

- Introduction
- Ideas:
  - Use mistakes to train subsequent models
  - Use many learners
  - Use simple algorithms (for now)
  - Weight sample contribution to error
  - Weight the learners
  - Change weights
- Conclusion