Ensembles and Model Evaluation

cs540 section 2
Louis Oliphant
oliphant@cs.wisc.edu

Announcements

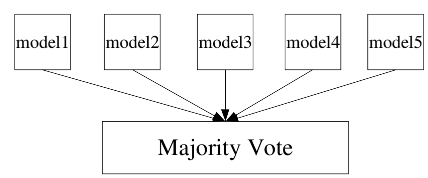
- Review Session
  - Tuesday, Nov 1st 4:30-5:30pm CS 1325
  - Come with quesions, no lecture prepared.
- Homework 3 due today
- Homework 2 returned today
  - Does NOT include the grade on the programming portion
    - still calculating that
    - Tournament is half over, we have the winners on the 7x7 standard board but still need to run on the previously "unseen" board

Two parts to Models

- Induction
  - Induce, Learn, Create, Make, Grow [a model]
- Inference
  - Infer, label, classify, deduce new examples with [a model]

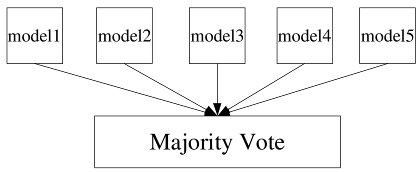
Two Heads are Better Than One

- induce N (say N=5) models from the training data
- Classify new examples by simple majority voting among the N models
- For the ensemble to mis-classify a new example, ***at least 3 of the 5 hypotheses have to mis-classify it***.



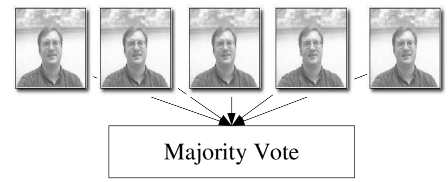
Ensembles

- Assume
  - Each hypothesis, h_i, has error rate of p
    - The probability that a randomly chosen example is misclassified.
  - Errors made by each hypothesis are independent
- With 5 hypothesis, if p=0.10 then the ensemble will mis-classify with a rate less than 0.01



Getting Independence

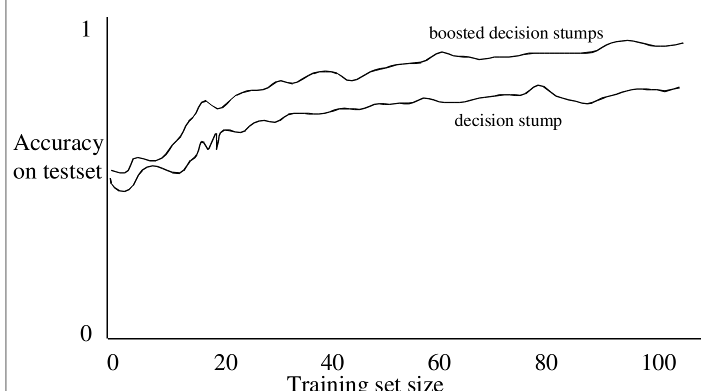
- What if each model were trained the same, on the same training set?
  - Would the models have independent errors?
- Boosting is a method to help in creating models that are different, thus independent, in mis-classification
- Different is Good! (at least when everybody else is wrong)



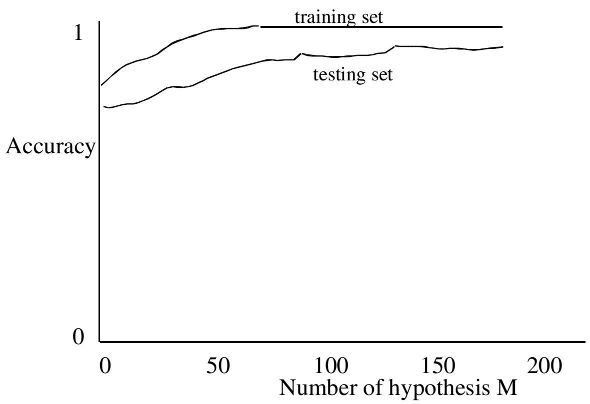
Boosting

- Each example in training set is weighted
  - Initial weight is 1
- Induce a model on training set, using weights
- Change weights
  - increase weight of examples in training set that are misclassified
  - decrease weight of examples in training set that are correctly classified
- Repeat until you have M models
- Classify using a weighted vote of the M models
- Understand the general idea of Adaboost algorithm (figure 18.10)

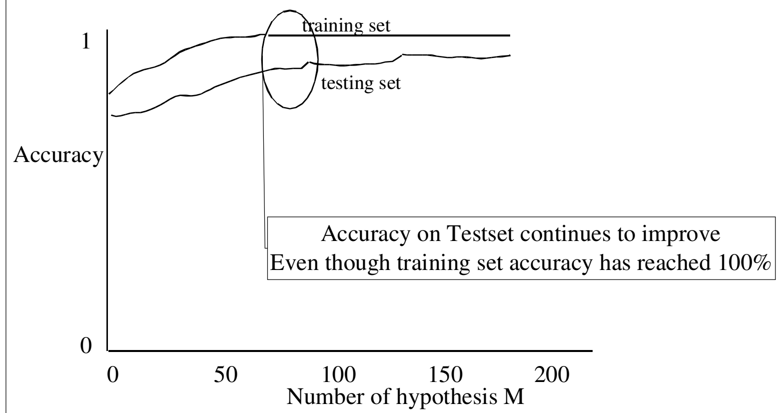
Performance of Ensembles
(learning curves)



Performance of Ensembles

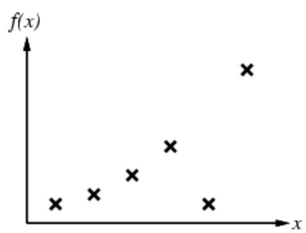


Performance of Ensembles



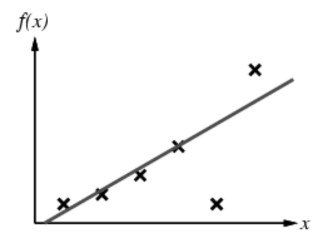
Inductive learning method

- Construct/adjust *h* to agree with *f* on training set
- (*h* is consistent if it agrees with *f* on all examples)
- E.g., curve fitting:



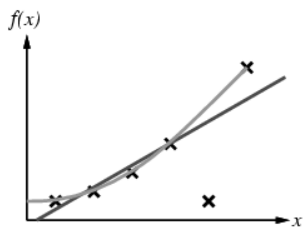
Inductive learning method

- Construct/adjust *h* to agree with *f* on training set
- (*h* is consistent if it agrees with *f* on all examples)
- E.g., curve fitting:



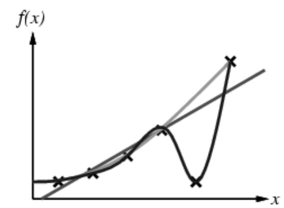
Inductive learning method

- Construct/adjust *h* to agree with *f* on training set
- (*h* is consistent if it agrees with *f* on all examples)
- E.g., curve fitting:



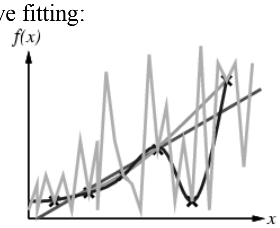
Inductive learning method

- Construct/adjust *h* to agree with *f* on training set
- (*h* is consistent if it agrees with *f* on all examples)
- E.g., curve fitting:



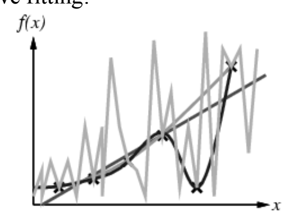
Inductive learning method

- Construct/adjust *h* to agree with *f* on training set
- (*h* is consistent if it agrees with *f* on all examples)
- E.g., curve fitting:



Inductive learning method

- Construct/adjust *h* to agree with *f* on training set
- (*h* is consistent if it agrees with *f* on all examples)
- E.g., curve fitting:



- Ockham's razor: prefer the simplest hypothesis consistent with data

Model Evaluation

- Given two models:
  - how do you decide which one is better for a given task (on a given dataset)?
  - Accuracy
  - Accuracy with cross-validation
  - Confusion Matrix
  - Recall, Precision

Model Evaluation

- Accuracy (inversely error rate)
  - What is the probability of labeling some new example correctly?
- Estimating Accuracy
  - Fraction of examples in some previously unseen dataset that are labeled correctly
  - Why is this just an estimate?

    The dataset may not be representative sample
    i.e. it is too easy or too hard

Reducing the Error in the Estimation

- N-Fold Cross Validation
  - For a given dataset split into N disjoint subsets

Dataset				
Set 1	Set 2	Set 3	Set 4	Set 5

  - Train on N-1 of the sets and test the accuracy of the left out set
  - Do this for each combination of train/test split (N possible ways)
  - Report the average accuracy of the N test set accuracies along with error bars (standard deviation)

N-Fold Cross Validation

Which Model would you choose? why?

- Model 1

0.78	0.72	0.77	0.73	0.80

  - average accuracy: 0.76
  - standard deviation: 0.03
- Model 2

0.62	0.88	0.70	0.81	0.77

  - average accuracy: 0.76
  - standard deviation: 0.10
- Standard Deviation

$$s = \sqrt{\text{var}} = \sqrt{\frac{\Sigma\left(X - \bar{X}\right)^2}{N-1}}$$

  - The standard deviation is defined as the average amount by which scores in a distribution differ from the mean

Confusion Matrix

- Imagine a model that predicts if a tumor is malignant or benign:
  - Is it just as bad to
    - incorrectly predict that a person has cancer when they don't
    - incorrectly predict that a person doesn't have cancer when they do
- When evaluating models we want to know what kind of errors they made – Create a Confusion Matrix of the models on the test set

Confusion Matrix

Actual

Predicted		pos	neg
	pos	TP	FP
	neg	FN	TN

TP – True Positives
FP – False Positives
FN – False Negatives
TN – True Negatives

Confusion Matrix

Model 1

Actual

Predicted		pos	neg
	pos	700	0
	neg	300	1000

Model 2

Actual

Predicted		pos	neg
	pos	1000	300
	neg	0	700

What is the accuracy of the two models?
Which model would you want diagnosing if your tumor were malignant or benign?

Skewed Data

- Hypothetical Dataset
  - Negatives – 500,000 examples
  - Positives – 100 examples
- Lots of real data is like this. Imagine The tumor scenario. Most people don't have cancer.
- Suppose you create a model that always guesses negative. What will your accuracy on the dataset be? 99.99% Wow, what a great model!
- But we want to get the positive examples right.
- Two metrics are commonly used when working with skewed data: precision and recall

Precision and Recall

- Recall – What fraction of the positive examples did your model find (predict positive)
  Recall=
- Precision – What fraction of the predicted positive examples were actually positive
  Precision=

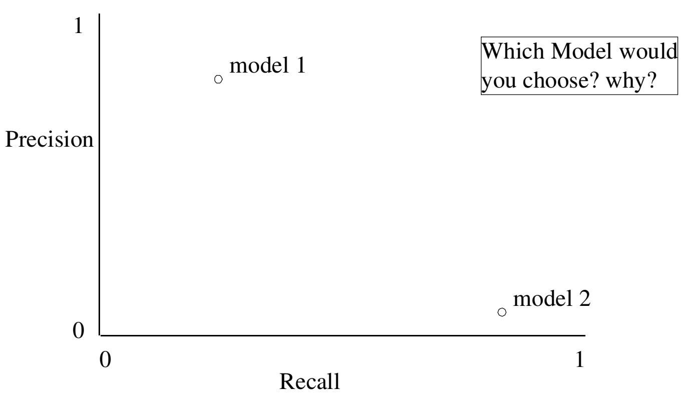
		Actual	
		pos	neg
Predicted	pos	TP	FP
	neg	FN	TN

Precision and Recall

- Recall – What fraction of the positive examples did your model find (predict positive)
  Recall= TP/(TP+FN)
- Precision – What fraction of the predicted positive examples were actually positive
  Precision= TP/(TP+FP)

		Actual	
		pos	neg
Predicted	pos	TP	FP
	neg	FN	TN

Recall and Precision "Space"



Conclusion

- Ensembles
- Ockam's Razor
- Accuracy
- N-Fold Cross Validation
- Confusion Matrix
- Recall and Precision