CSDS 440: Machine Learning

Soumya Ray (he/him, sray@case.edu)

Olin 516

Office hours T, Th 11:15-11:45 or by appointment

Announcements

- Test 1 next Thursday 9/26, in class, 30-45 minutes, closed book/notes
  - Topics: everything up to and including decision trees
  - Remember to review probability and statistics

Today

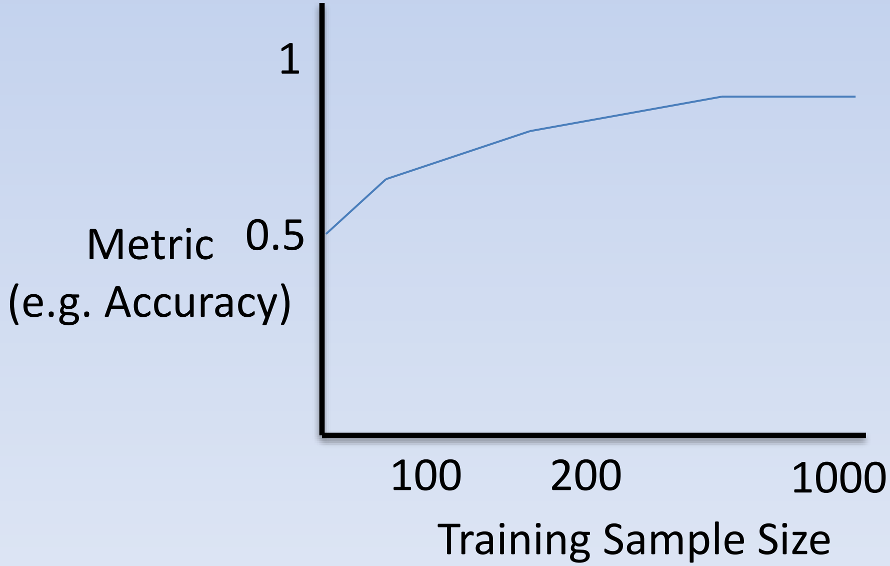
- Metrics
- Comparing Learning Algorithms

Beyond point estimates

- Everything above is a “point estimate”
- Because they will be computed on the basis of a sample, we can also compute variance estimates for each quantity
- Important to show “stability” of solutions, and when comparing across algorithms (later)

Learning Curves

- Often useful to plot each metric as a function of training sample size
- Provides insight into how many examples the algorithm needs to become effective



Metrics with Confidence Measures

- Many learning algorithms can produce models that can provide estimates of how *confident* they are about a prediction

- Example: Pruned Decision Trees

Metrics with Confidence Measures

	True Class	Confidence On +
Example 1	+	0.9
Example 2	–	0.8
Example 3	+	0.4
Example 4	–	0.3

- We can create *multiple classifiers* by *thresholding* the confidence

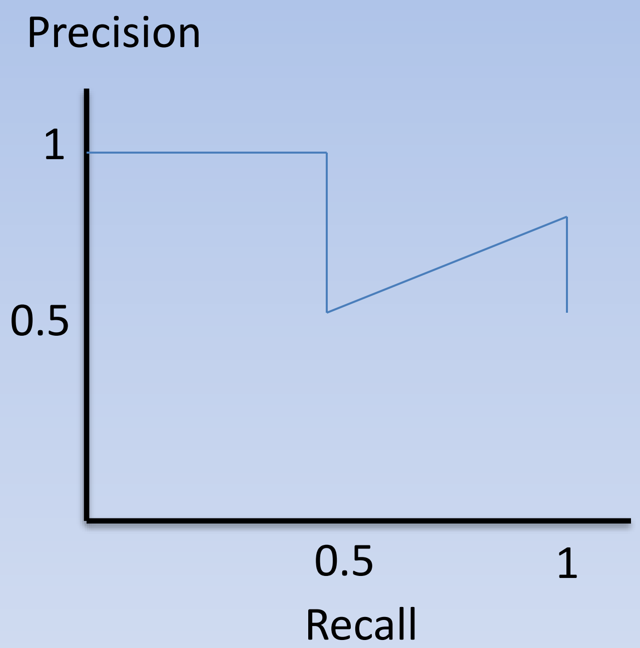
- In this case, we can plot Precision-Recall (PR) and Receiver Operating Characteristic (ROC) graphs tracking *all* of the classifiers

Precision-Recall graphs

	True Class	Confidence On +	Recall (x axis)	Precision (y axis)
Example 1	+	0.9		
Example 2	–	0.8		
Example 3	+	0.4		
Example 4	–	0.3		

Precision-Recall graphs

	True Class	Confidence On +	Recall (x axis)	Precision (y axis)
Example 1	+	0.9	0.5	1
Example 2	–	0.8	0.5	0.5
Example 3	+	0.4	1	0.67
Example 4	–	0.3	1	0.5

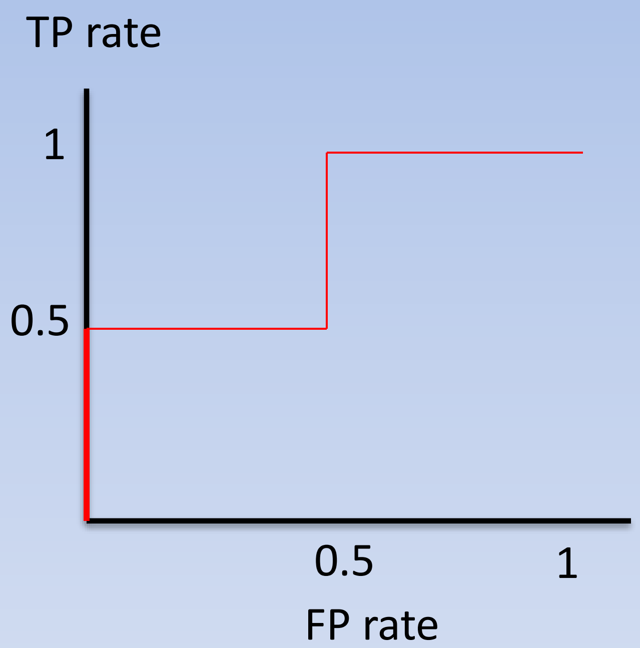


ROC graphs

	True Class	Confidence On +	FP Rate (1-Spec.) (x axis)	Sens./Recall (y axis)
Example 1	+	0.9		
Example 2	–	0.8		
Example 3	+	0.4		
Example 4	–	0.3		

ROC graphs

	True Class	Confidence On +	FP Rate (x axis)	Sens./Recall (y axis)
Example 1	+	0.9	0	0.5
Example 2	–	0.8	0.5	0.5
Example 3	+	0.4	0.5	1
Example 4	–	0.3	1	1



Properties of ROC graphs

- Random guessing is a diagonal line
  - Also majority class classifier
  - If your classifier is any good its ROC must lie above the diagonal
- Monotonically increasing
- Often use “AUC”/ “AROC” as comparison statistic (later)
- Can be misleading if class distribution is too skewed (use PR graphs instead)

Comparing Learning Algorithms

Key Issue #1

- Suppose we collect some test data from a binary classification problem and evaluate a classifier. The accuracy is x.

- Then we (or someone else) repeats the experiment with another set of test data from the same problem, collected independently of the first set.
  - What can we say about the accuracy in this case?

Key Issue #2.1

- Suppose we have *two classifiers* A and B. We measure their accuracies on a test set, they are x and y and $x > y$. Does this mean A is better than B for this problem?

- What if we (or someone else) re-did the experiment with another test set? Would we still find $x > y$?

Key Issue #2.2

- Suppose we have two *learning algorithms* A and B. We measure their accuracies on a problem, they are x and y and $x > y$. Does this mean A is better than B for this problem?

Main Idea

- Earlier we saw how to calculate various metrics for a classifier

- We will always calculate these metrics on the basis of a *small, finite sample*

- What can we say about the *true value* of the metric from our estimates?

Data Distribution

- Assume there is an unknown, underlying probability distribution, D, from which *unlabeled* examples (x) are being sampled with replacement

- Examples are I.I.D.

- D is unknown, but fixed

Sample Error Rate

- The fraction of examples in our test sample on which the learned classifier disagrees with the target concept

$$e_S = \frac{1}{n} \sum_{x \in S} \delta(y_x, \hat{y}_x)$$

$$\delta(y_x, \hat{y}_x) = 1 \text{ if } y_x \neq \hat{y}_x, 0 \text{ else}$$

$$n = \text{sample size}$$

True Error Rate

- The probability that the learned classifier will make a mistake *on a random example drawn from D*

$$e_D = \Pr_{x \sim D}(y_x \neq \hat{y}_x)$$

- This is what we *really* want to know

Issue #1 Problem Setup

- A test set of size n is drawn from an underlying unknown data distribution D. A learned classifier is evaluated on this sample.
- Sample error rate: $e_S = \frac{1}{n}\sum_x \delta(y_x, \hat{y}_x)$
- True error rate: $e_D = \Pr_{x \sim D}(y_x \neq \hat{y}_x)$
- Question: How are e_S and e_D related?

Sampling Distribution

- Suppose we perform a random experiment lots of times and record the outcome
- Call the random variable associated with the outcome O
- Suppose we then plot a frequency histogram of O
  - For each value of O, record the number of times we saw it during our experiments
- This is the sampling distribution of O

Sampling Distribution of Number of Errors

- Let R be a r.v. denoting the *number* of errors in an evaluation experiment

$$r = \sum_{x \in S} \delta(y_x, \hat{y}_x)$$

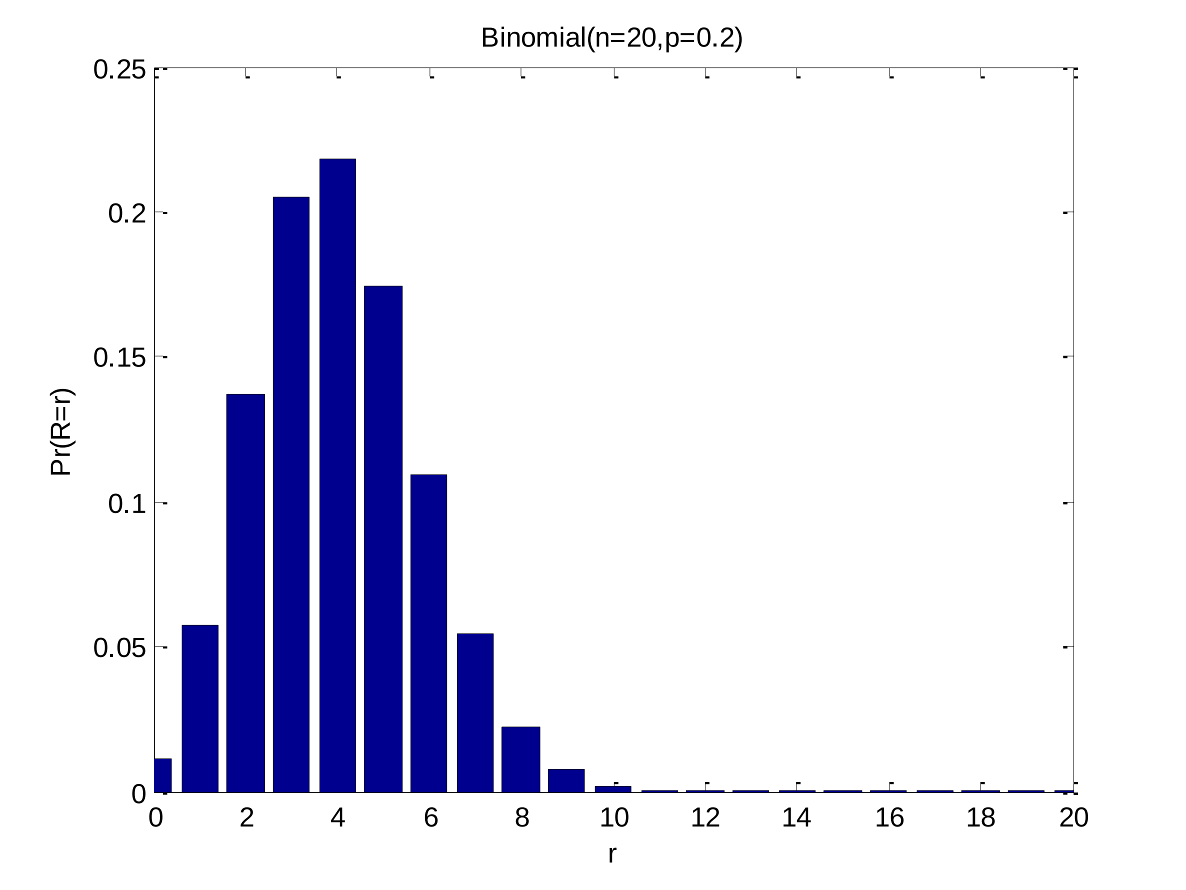
- What is the sampling distribution of R?

Sampling Distribution of R

- Suppose we run k experiments with test samples of size n
- In the i^{th} experiment our learned classifier makes $R=r_i$ errors
- We plot a frequency histogram of R
- What does this look like as k gets large?

Sampling Distribution of R

- It is a Binomial distribution



$$B(R=r;n,p)=\binom{n}{r}p^{r}(1-p)^{n-r}$$

Why?

- Let us imagine we have a coin that shows heads with probability e_D
- We flip it n times and count the number of heads. Repeat and plot a frequency histogram.
- You get a binomial distribution with parameters e_D and n

Why?

- For binary classification with i.i.d examples, each example is like a trial where our classifier has probability of failure e_D

- This is analogous to the situation where you have a coin that shows "heads" with probability e_D

- So if you plot the distribution of the number of errors ("heads"), it will also be a Binomial distribution with parameters n and e_D

Useful Binomial Facts

- Expectation of a Binomial random variable R with distribution $B(n,e_D)$

$$E(R) = ne_D$$

- Variance of a Binomial random variable with distribution $B(n,e_D)$

$$V(R) = ne_D(1 - e_D)$$

Parameter Estimation

- Notice that in this case, we are working with a distribution whose parameters are unknown
  - We are trying to *estimate* e_D , given r and n

- Suppose we only did a single experiment with n examples and observed r errors
  - What is a good estimate of e_D?

Parameter Estimation

- It is $e_S = r/n$. Why?
- This is the estimate that, under the Binomial distribution, *maximizes the likelihood of the observed number of errors*:

$$\hat{e}_D = \arg\max_p B(R = r; n, p) = e_S = \frac{r}{n}$$

- Called the Maximum Likelihood Estimate, or MLE

Estimation Bias

- The estimation bias of an estimator Y for a parameter p is $E(Y)$-p
- If an estimator has zero bias then the average estimate will converge to the true value
- The MLE has asymptotically zero estimation bias