CSCE 478/878 Lecture 4: Experimental Design and Analysis

Stephen Scott

(Adapted from Ethem Alpaydin and Tom Mitchell)

sscott@cse.unl.edu

Introduction

In Homework 1, you are (supposedly)

1. Choosing a data set
2. Extracting a test set of size > 30
3. Building a tree on the training set
4. Testing on the test set
5. Reporting the accuracy

Does the reported accuracy exactly match the generalization performance of the tree?

If a tree has error 10% and an ANN has error 11%, is the tree absolutely better?

- Why or why not?

How about the algorithms in general?

Outline

- Goals of performance evaluation
- Estimating error and confidence intervals
- Paired t tests and cross-validation to compare learning algorithms
- Other performance measures
  - Confusion matrices
  - ROC analysis
  - Precision-recall curves

Setting Goals

- Before setting up an experiment, need to understand exactly what the goal is
  - Estimate the generalization performance of a hypothesis
  - Tuning a learning algorithm's parameters
  - Comparing two learning algorithms on a specific task
  - Etc.
- Will never be able to answer the question with 100% certainty
  - Due to variances in training set selection, test set selection, etc.
  - Will choose an *estimator* for the quantity in question, determine the probability distribution of the estimator, and bound the probability that the estimator is way off
  - Estimator needs to work regardless of distribution of training/testing data

Setting Goals (cont'd)

- Need to note that, in addition to statistical variations, what we determine is limited to the application that we are studying
  - E.g., if naïve Bayes better than ID3 on spam filtering, that means nothing about face recognition
- In planning experiments, need to ensure that training data not used for evaluation
  - I.e., *don't test on the training set!*
  - Will bias the performance estimator
  - Also holds for *validation set* used to prune DT, tune parameters, etc.
    - Validation set serves as part of training set, but not used for model building

Types of Error

- For now, focus on straightforward, 0/1 *classification error*
- For hypothesis h, recall the two types of classification error from Chapter 2:
  - *Empirical error* (or *sample error*) is fraction of set $\mathcal{V}$ that h gets wrong:

$$error_{\mathcal{V}}(h) \equiv \frac{1}{|\mathcal{V}|} \sum_{x \in \mathcal{V}} \delta(C(x) \neq h(x)) ,$$

  where $\delta(C(x) \neq h(x))$ is 1 if $C(x) \neq h(x)$, and 0 otherwise
  - *Generalization error* (or *true error*) is probability that a new, randomly selected, instance is misclassified by h

$$error_{\mathcal{D}}(h) \equiv \Pr_{x \in \mathcal{D}}[C(x) \neq h(x)] ,$$

  where $\mathcal{D}$ is probability distribution instances are drawn from
- Why do we care about $error_{\mathcal{V}}(h)$?

Estimating True Error

- *Bias*: If $\mathcal{T}$ is training set, $error_{\mathcal{T}}(h)$ is optimistically biased

$$bias \equiv E[error_{\mathcal{T}}(h)] - error_{\mathcal{D}}(h)$$

  For unbiased estimate ($bias = 0$), h and $\mathcal{V}$ must be chosen independently ⇒ *Don't test on training set!* (Don't confuse with inductive bias!)

- *Variance*: Even with unbiased $\mathcal{V}$, $error_{\mathcal{V}}(h)$ may still *vary* from $error_{\mathcal{D}}(h)$

Nebraska Lincoln
CSCE 478/878 Lecture 4: Experimental Design and Analysis
Stephen Scott
Introduction
Outline
Goals
Estimating Error
Types of Error
Estimating Error
Confidence Intervals
Comparing Learning Algorithms
Other Performance Measures

Estimating True Error (cont'd)

Experiment:

1. Choose sample $\mathcal{V}$ of size N according to distribution $\mathcal{D}$
2. Measure $error_{\mathcal{V}}(h)$

$error_{\mathcal{V}}(h)$ is a random variable (i.e., result of an experiment)

$error_{\mathcal{V}}(h)$ is an *unbiased estimator* for $error_{\mathcal{D}}(h)$

Given observed $error_{\mathcal{V}}(h)$, what can we conclude about $error_{\mathcal{D}}(h)$?

Nebraska Lincoln
CSCE 478/878 Lecture 4: Experimental Design and Analysis
Stephen Scott
Introduction
Outline
Goals
Estimating Error
Types of Error
Estimating Error
Confidence Intervals
Comparing Learning Algorithms
Other Performance Measures

Confidence Intervals

If

- $\mathcal{V}$ contains N examples, drawn independently of h and each other
- $N \geq 30$

Then with approximately 95% probability, $error_{\mathcal{D}}(h)$ lies in

$$error_{\mathcal{V}}(h) \pm 1.96\sqrt{\frac{error_{\mathcal{V}}(h)(1 - error_{\mathcal{V}}(h))}{N}}$$

E.g. hypothesis h misclassifies 12 of the 40 examples in test set $\mathcal{V}$:

$$error_{\mathcal{V}}(h) = \frac{12}{40} = 0.30$$

Then with approx. 95% confidence, $error_{\mathcal{D}}(h) \in [0.158, 0.442]$

Nebraska Lincoln
CSCE 478/878 Lecture 4: Experimental Design and Analysis
Stephen Scott
Introduction
Outline
Goals
Estimating Error
Types of Error
Estimating Error
Confidence Intervals
Comparing Learning Algorithms
Other Performance Measures

Confidence Intervals (cont'd)

If

- $\mathcal{V}$ contains N examples, drawn independently of h and each other
- $N \geq 30$

Then with approximately c% probability, $error_{\mathcal{D}}(h)$ lies in

$$error_{\mathcal{V}}(h) \pm z_c\sqrt{\frac{error_{\mathcal{V}}(h)(1 - error_{\mathcal{V}}(h))}{N}}$$

$N\%$:	50%	68%	80%	90%	95%	98%	99%
z_c:	0.67	1.00	1.28	1.64	1.96	2.33	2.58

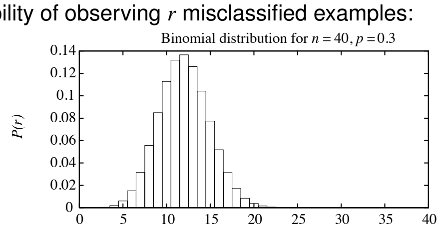
Why?

Nebraska Lincoln
CSCE 478/878 Lecture 4: Experimental Design and Analysis
Stephen Scott
Introduction
Outline
Goals
Estimating Error
Types of Error
Estimating Error
Confidence Intervals
Comparing Learning Algorithms
Other Performance Measures

$error_{\mathcal{V}}(h)$ is a Random Variable

Repeatedly run the experiment, each with different randomly drawn $\mathcal{V}$ (each of size N)

Probability of observing r misclassified examples:



$$P(r) = \binom{N}{r} error_{\mathcal{D}}(h)^r (1 - error_{\mathcal{D}}(h))^{N-r}$$

I.e., let $error_{\mathcal{D}}(h)$ be probability of heads in biased coin, then $P(r)$ = prob. of getting r heads out of N flips

Nebraska Lincoln
CSCE 478/878 Lecture 4: Experimental Design and Analysis
Stephen Scott
Introduction
Outline
Goals
Estimating Error
Types of Error
Estimating Error
Confidence Intervals
Comparing Learning Algorithms
Other Performance Measures

Binomial Probability Distribution

$$P(r) = \binom{N}{r} p^r (1-p)^{N-r} = \frac{N!}{r!(N-r)!} p^r (1-p)^{N-r}$$

Probability $P(r)$ of r heads in N coin flips, if $p = \Pr(heads)$

- Expected, or mean value of X, $E[X]$ (= # heads on N flips = # mistakes on N test exs), is

$$E[X] \equiv \sum_{i=0}^{N} iP(i) = Np = N \cdot error_{\mathcal{D}}(h)$$

- Variance of X is

$$Var(X) \equiv E[(X - E[X])^2] = Np(1-p)$$

- Standard deviation of X, σ_X, is

$$\sigma_X \equiv \sqrt{E[(X - E[X])^2]} = \sqrt{Np(1-p)}$$

Approximate Binomial Dist. with Normal

$error_{\mathcal{V}}(h) = r/N$ is binomially distributed, with

- mean $\mu_{error_{\mathcal{V}}(h)} = error_{\mathcal{D}}(h)$ (i.e., unbiased est.)
- standard deviation $\sigma_{error_{\mathcal{V}}(h)}$

$$\sigma_{error_{\mathcal{V}}(h)} = \sqrt{\frac{error_{\mathcal{D}}(h)(1 - error_{\mathcal{D}}(h))}{N}}$$

(increasing N decreases variance)

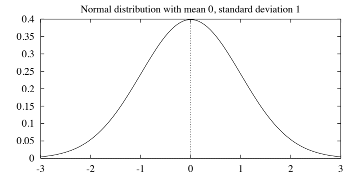
Want to compute confidence interval = interval centered at $error_{\mathcal{D}}(h)$ containing c% of the weight under the distribution

Approximate binomial by *normal* (Gaussian) dist:

- mean $\mu_{error_{\mathcal{V}}(h)} = error_{\mathcal{D}}(h)$
- standard deviation $\sigma_{error_{\mathcal{V}}(h)}$

$$\sigma_{error_{\mathcal{V}}(h)} \approx \sqrt{\frac{error_{\mathcal{V}}(h)(1 - error_{\mathcal{V}}(h))}{N}}$$

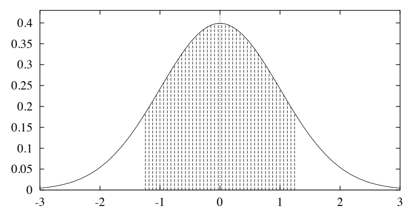
Normal Probability Distribution



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

- The probability that X will fall into the interval (a, b) is given by $\int_a^b p(x)\,dx$
- Expected, or mean value of X, $E[X]$, is $E[X] = \mu$
- Variance is $Var(X) = \sigma^2$, standard deviation is $\sigma_X = \sigma$

Normal Probability Distribution (cont'd)



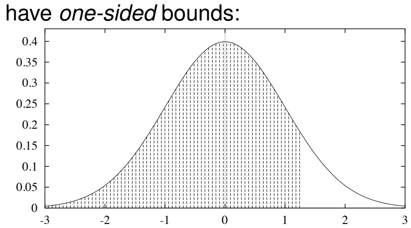
80% of area (probability) lies in $\mu \pm 1.28\sigma$

c% of area (probability) lies in $\mu \pm z_c\,\sigma$

c%:	50%	68%	80%	90%	95%	98%	99%
z_c:	0.67	1.00	1.28	1.64	1.96	2.33	2.58

Normal Probability Distribution (cont'd)

Can also have *one-sided* bounds:



c% of area lies $< \mu + z'_c\,\sigma$ or $> \mu - z'_c\sigma$, where $z'_c = z_{100-(100-c)/2}$

c%:	50%	68%	80%	90%	95%	98%	99%
z'_c:	0.0	0.47	0.84	1.28	1.64	2.05	2.33

Confidence Intervals Revisited

If $\mathcal{V}$ contains $N \geq 30$ examples, indep. of h and each other

Then with approximately 95% probability, $error_{\mathcal{V}}(h)$ lies in

$$error_{\mathcal{D}}(h) \pm 1.96\sqrt{\frac{error_{\mathcal{D}}(h)(1 - error_{\mathcal{D}}(h))}{N}}$$

Equivalently, $error_{\mathcal{D}}(h)$ lies in

$$error_{\mathcal{V}}(h) \pm 1.96\sqrt{\frac{error_{\mathcal{D}}(h)(1 - error_{\mathcal{D}}(h))}{N}}$$

which is approximately

$$error_{\mathcal{V}}(h) \pm 1.96\sqrt{\frac{error_{\mathcal{V}}(h)(1 - error_{\mathcal{V}}(h))}{N}}$$

(*One-sided bounds yield upper or lower error bounds*)

Central Limit Theorem

How can we justify approximation?

Consider set of iid random variables $Y_1, \ldots, Y_N$, all from *arbitrary* probability distribution with mean μ and finite variance σ^2. Define sample mean $\bar{Y} \equiv (1/N)\sum_{i=1}^{n} Y_i$

$\bar{Y}$ is itself a random variable, i.e., result of an experiment (e.g., $error_S(h) = r/N$)

Central Limit Theorem: As $N \to \infty$, the distribution governing $\bar{Y}$ approaches normal distribution with mean μ and variance σ^2/N

Thus the distribution of $error_S(h)$ is approximately normal for large N, and its expected value is $error_{\mathcal{D}}(h)$

(Rule of thumb: $N \geq 30$ when estimator's distribution is binomial; might need to be larger for other distributions)

Calculating Confidence Intervals

1. Pick parameter to estimate: $error_{\mathcal{D}}(h)$
2. Choose an estimator: $error_{\mathcal{V}}(h)$
3. Determine probability distribution that governs estimator: $error_{\mathcal{V}}(h)$ governed by binomial distribution, approximated by normal when $N \geq 30$
4. Find interval (L, U) such that c% of probability mass falls in the interval
   - Could have $L = -\infty$ or $U = \infty$
   - Use table of z_c or z'_c values (if distribution normal)

Comparing Learning Algorithms

- What if we want to compare two learning algorithms L^1 and L^2 (e.g., ID3 vs k-nearest neighbor) on a specific application?
- Insufficient to simply compare error rates on a single test set
- Use *K-fold cross validation* and a *paired t test*

K-Fold Cross Validation

1. Partition data set $\mathcal{X}$ into K equal-sized subsets $\mathcal{X}_1, \mathcal{X}_2, \ldots, \mathcal{X}_K$, where $|\mathcal{X}_i| \geq 30$
2. For i from 1 to K, do
   (Use $\mathcal{X}_i$ for testing, and rest for training)
   1. $\mathcal{V}_i = \mathcal{X}_i$
   2. $\mathcal{T}_i = \mathcal{X} \setminus \mathcal{X}_i$
   3. Train learning algorithm L^1 on $\mathcal{V}_i$ to get h^1
   4. Train learning algorithm L^2 on $\mathcal{V}_i$ to get h^2
   5. Let p_i^j be error of h^j on test set $\mathcal{V}_i$
   6. $p_i = p_i^1 - p_i^2$
3. Error difference estimate $p = (1/K) \sum_i^K p_i$

K-Fold Cross Validation (cont'd)

- Now want to determine confidence that $p < 0$

$\Rightarrow$ Confidence that L^1 is better than L^2 on learning task

- Use *one-sided test*, with confidence derived from *student's t distribution* with $K - 1$ *degrees of freedom*
- With approximately c% probability, true difference of expected error between L^1 and L^2 is at most

$$p + t_{c,K-1}\, s_p$$

where

$$s_p \equiv \sqrt{\frac{1}{K(K-1)} \sum_{i=1}^{K} (p_i - p)^2}$$

Student's t Distribution (One-Sided Test)

df	0.600	0.700	0.800	0.900	0.950	0.975	0.990	0.995
1	0.325	0.727	1.376	3.078	6.314	12.706	31.821	63.657
2	0.289	0.617	1.061	1.886	2.920	4.303	6.965	9.925
3	0.277	0.584	0.978	1.638	2.353	3.182	4.541	5.841
4	0.271	0.569	0.941	1.533	2.132	2.776	3.747	4.604
5	0.267	0.559	0.920	1.476	2.015	2.571	3.365	4.032
6	0.265	0.553	0.906	1.440	1.943	2.447	3.143	3.707
7	0.263	0.549	0.896	1.415	1.895	2.365	2.998	3.499
8	0.262	0.546	0.889	1.397	1.860	2.306	2.896	3.355
9	0.261	0.543	0.883	1.383	1.833	2.262	2.821	3.250
10	0.260	0.542	0.879	1.372	1.812	2.228	2.764	3.169
11	0.260	0.540	0.876	1.363	1.796	2.201	2.718	3.106
12	0.259	0.539	0.873	1.356	1.782	2.179	2.681	3.055
13	0.259	0.538	0.870	1.350	1.771	2.160	2.650	3.012

If $p + t_{c,K-1}\, s_p < 0$ our assertion that L^1 has less error than L^2 is supported with confidence c

So if K-fold CV used, compute p, look up $t_{c,K-1}$ and check if $p < -t_{c,K-1}\, s_p$

One-sided test; says nothing about L^2 over L^1

Caveat

- Say you want to show that learning algorithm L^1 performs better than algorithms L^2, L^3, L^4, L^5
- If you use K-fold CV to show superior performance of L^1 over each of $L^2, \ldots, L^5$ at 95% confidence, there's a 5% chance each one is wrong

$\Rightarrow$ There's a 20% chance that at least one is wrong

$\Rightarrow$ Our overall confidence is only 80%

- Need to account for this
- Or, use other statistical tests to analyze multiple algorithms

More Specific Performance Measures

- So far, we've looked at a single error rate to compare hypotheses/learning algorithms/etc.
- This may not tell the whole story:
  - 1000 test examples: 20 positive, 980 negative
  - h^1 gets 2/20 pos correct, 965/980 neg correct, for accuracy of $(2 + 965)/(20 + 980) = 0.967$
  - Pretty impressive, except that always predicting negative yields accuracy $= 0.980$
  - Would we rather have h^2, which gets 19/20 pos correct and 930/980 neg, for accuracy $= 0.949$?
  - Depends on how important the positives are, i.e., frequency in practice and/or cost (e.g., cancer diagnosis)

Nebraska Lincoln
CSCE 478/878 Lecture 4: Experimental Design and Analysis
Stephen Scott
Introduction
Outline
Goals
Estimating Error
Comparing Learning Algorithms
Other Performance Measures
Confusion Matrices
ROC Curves
Precision-Recall Curves

Confusion Matrices

Break down error into type: true positive, etc.

	Predicted Class		
True Class	Positive	Negative	Total
Positive	tp : true positive	fn : false negative	p
Negative	fp : false positive	tn : true negative	n
Total	p'	n'	N

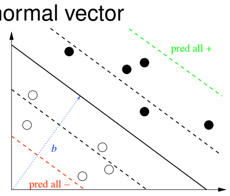
Generalizes to multiple classes

Allows one to quickly assess which classes are missed the most, and into what other class

Nebraska Lincoln
CSCE 478/878 Lecture 4: Experimental Design and Analysis
Stephen Scott
Introduction
Outline
Goals
Estimating Error
Comparing Learning Algorithms
Other Performance Measures
Confusion Matrices
ROC Curves
Precision-Recall Curves

ROC Curves

- Consider an ANN or SVM
- Normally threshold at 0, but what if we changed it?
- Keeping weight vector constant while changing threshold = holding hyperplane's slope fixed while moving along its normal vector



- I.e., get a *set* of classifiers, one per labeling of test set
- Similar situation with any classifier with confidence value, e.g., probability-based

Nebraska Lincoln
CSCE 478/878 Lecture 4: Experimental Design and Analysis
Stephen Scott
Introduction
Outline
Goals
Estimating Error
Comparing Learning Algorithms
Other Performance Measures
Confusion Matrices
ROC Curves
Precision-Recall Curves

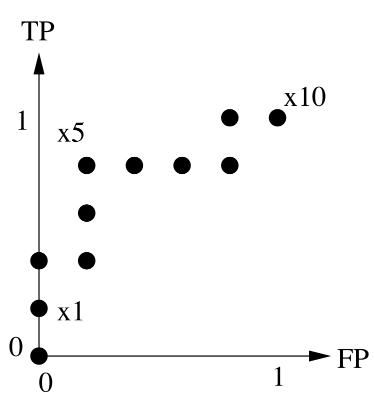
ROC Curves
Plotting tp versus fp

- Consider the "always $-$" hyp. What is fp? What is tp? What about the "always $+$" hyp?
- In between the extremes, we plot TP versus FP by sorting the test examples by the confidence values

Ex	Confidence	label	Ex	Confidence	label
x_1	169.752	$+$	x_6	-12.640	$-$
x_2	109.200	$+$	x_7	-29.124	$-$
x_3	19.210	$-$	x_8	-83.222	$-$
x_4	1.905	$+$	x_9	-91.554	$+$
x_5	-2.75	$+$	x_{10}	-128.212	$-$

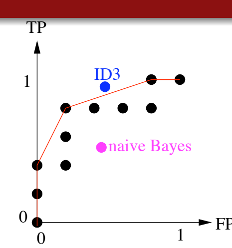
Nebraska Lincoln
CSCE 478/878 Lecture 4: Experimental Design and Analysis
Stephen Scott
Introduction
Outline
Goals
Estimating Error
Comparing Learning Algorithms
Other Performance Measures
Confusion Matrices
ROC Curves
Precision-Recall Curves

ROC Curves
Plotting tp versus fp (cont'd)



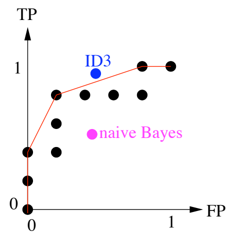
Nebraska Lincoln
CSCE 478/878 Lecture 4: Experimental Design and Analysis
Stephen Scott
Introduction
Outline
Goals
Estimating Error
Comparing Learning Algorithms
Other Performance Measures
Confusion Matrices
ROC Curves
Precision-Recall Curves

ROC Curves
Convex Hull



- The *convex hull* of the ROC curve yields a collection of classifiers, each optimal under different conditions
  - If FP cost = FN cost, then draw a line with slope $|N|/|P|$ at $(0, 1)$ and drag it towards convex hull until you touch it; that's your operating point
  - Can use as a classifier any part of the hull since can randomly select between two classifiers

ROC Curves
Convex Hull



- Can also compare curves against "single-point" classifiers when no curves
  - In plot, ID3 better than our SVM iff negatives scarce; nB never better

ROC Curves
Miscellany

- What is the worst possible ROC curve?
- One metric for measuring a curve's goodness: *area under curve* (AUC):

$$\frac{\sum_{x_+ \in P} \sum_{x_- \in N} I(h(x_+) > h(x_-))}{|P|\,|N|}$$

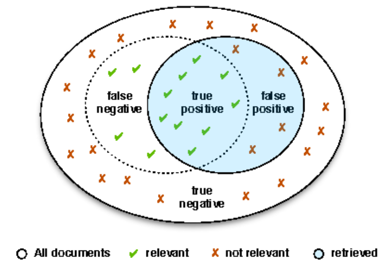
  i.e., rank all examples by confidence in "+" prediction, count the number of times a positively-labeled example (from P) is ranked above a negatively-labeled one (from N), then normalize
  - What is the best value?
  - Distribution approximately normal if $|P|, |N| > 10$, so can find confidence intervals
  - Catching on as a better scalar measure of performance than error rate
- ROC analysis possible (though tricky) with multi-class problems

ROC Curves
Miscellany (cont'd)

- Can use ROC curve to modify classifiers, e.g., re-label decision trees
- What does "ROC" stand for?
  - "Receiver Operating Characteristic" from signal detection theory, where binary signals are corrupted by noise
  - Use plots to determine how to set threshold to determine presence of signal
  - Threshold too high: miss true hits (tp low), too low: too many false alarms (fp high)
- Alternative to ROC: *cost curves*

Precision-Recall Curves

Consider information retrieval task, e.g., web search

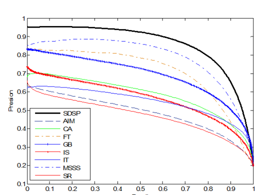


precision $= tp/p'$ = fraction of retrieved that are positive

recall $= tp/p$ = fraction of positives retrieved

Precision-Recall Curves (cont'd)

As with ROC, can vary threshold to trade off precision against recall



Can compare curves based on containment

Use F_β-measure to combine at a specific point, where β weights precision vs recall:

$$F_\beta \equiv (1+\beta^2)\frac{precision \cdot recall}{(\beta^2 \cdot precision) + recall}$$