Evaluating Machine-Learning Methods (Part 2)

Mark Craven and David Page
Computer Sciences 760
Spring 2018

www.biostat.wisc.edu/~craven/cs760/

Some of the slides in these lectures have been adapted/borrowed from materials developed by Tom Dietterich, Pedro Domingos, Tom Mitchell, David Page, and Jude Shavlik

Goals for the lecture

you should understand the following concepts

- confidence intervals for error
- pairwise *t*-tests for comparing learning systems
- scatter plots for comparing learning systems
- lesion studies
- model selection
- validation (tuning) sets
- internal cross validation

Confidence intervals on error

Given the observed error (accuracy) of a model over a limited sample of data, how well does this error characterize its accuracy over additional instances?

Suppose we have

- a learned model h
- a test set S containing n instances drawn independently of one another and independent of h
- $n \geq 30$
- h makes r errors over the n instances

our best estimate of the error of h is

$$error_S(h) = \frac{r}{n}$$

Confidence intervals on error

With approximately C% probability, the true error lies in the interval

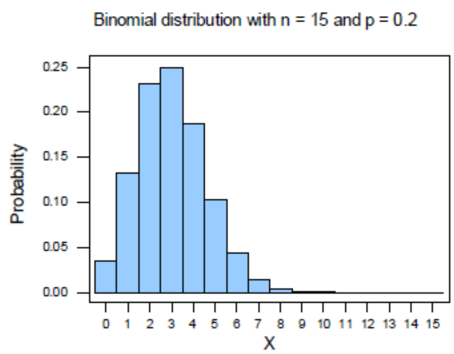
$$error_S(h) \pm z_C \sqrt{\frac{error_S(h)(1 - error_S(h))}{n}}$$

where z_C is a constant that depends on C (e.g. for 95% confidence, z_C =1.96)

Confidence intervals on error

How did we get this?

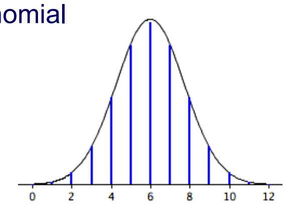
1. Our estimate of the error follows a binomial distribution given by n and p (the true error rate over the data distribution)



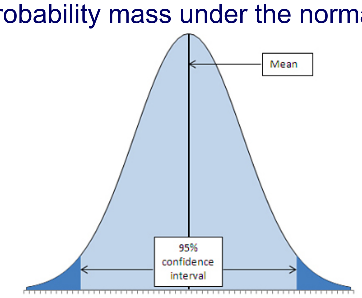
2. Most common way to determine a binomial confidence interval is to use the *normal approximation* (although can calculate exact intervals if n is not too large)

Confidence intervals on error

2. When $n \geq 30$, and p is not too extreme, the normal distribution is a good approximation to the binomial



3. We can determine the C% confidence interval by determining what bounds contain C% of the probability mass under the normal



Alternative approach: confidence intervals using bootstrapping

- *bootstrap sample*: given n examples in data set, randomly, uniformly, independently draw n examples with replacement

- repeat 1000 (or 10,000) times:
  - draw bootstrap sample
  - measure error on bootstrap sample
  - for 95% confidence interval, lower (upper) bound is set such that 2.5% of runs yield lower (higher) error

Comparing learning systems

How can we determine if one learning system provides better performance than another

- for a particular task?
- across a set of tasks / data sets?

Motivating example

	Accuracies on test sets				
System A:	80%	50	75	…	99
System B:	79	49	74	…	98
δ :	+1	+1	+1	…	+1

- Mean accuracy for System A is better, but the standard deviations for the two clearly overlap
- Notice that System A is always better than System B

Comparing systems using a paired t test

- consider δ's as observed values of a set of i.i.d. random variables

- *null hypothesis*: the 2 learning systems have the same accuracy
- *alternative hypothesis*: one of the systems is more accurate than the other

- hypothesis test:
  - use paired t-test do determine probability p that mean of δ's would arise from null hypothesis
  - if p is sufficiently small (typically < 0.05) then reject the null hypothesis

Comparing systems using a paired t test

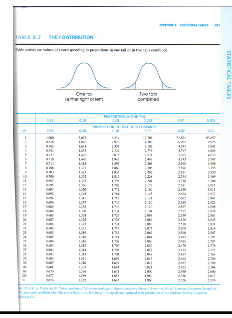
1. calculate the sample mean

$$\bar{\delta} = \frac{1}{n}\sum_{i=1}^{n}\delta_i$$

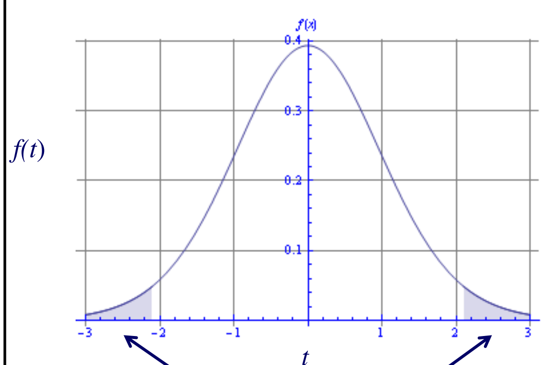
2. calculate the t statistic

$$t = \frac{\bar{\delta}}{\sqrt{\frac{1}{n(n-1)}\sum_{i=1}^{n}(\delta_i - \bar{\delta})^2}}$$

3. determine the corresponding p-value, by looking up t in a table of values for the Student's t-distribution with $n-1$ degrees of freedom



Comparing systems using a paired t test



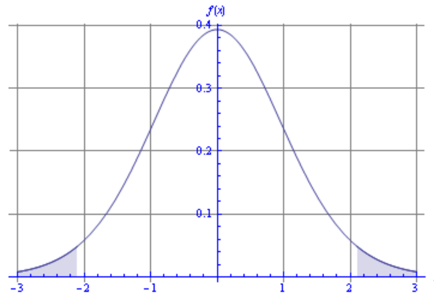
for a two-tailed test, the p-value represents the probability mass in these two regions

The null distribution of our t statistic looks like this

The p-value indicates how far out in a tail our t statistic is

If the p-value is sufficiently small, we reject the null hypothesis, since it is unlikely we'd get such a t by chance

Why do we use a two-tailed test?



- a two-tailed test asks the question: is the accuracy of the two systems different
- a one-tailed test asks the question: is system A better than system B
- a priori, we don't know which learning system will be more accurate (if there is a difference) – we want to allow that either one might be

Comments on hypothesis testing to compare learning systems

- the paired *t*-test can be used to compare two <u>learning systems</u>
- other tests (e.g. McNemar's χ^2 test) can be used to compare two <u>learned models</u>
- a statistically significant difference is not necessarily a large-magnitude difference

Scatter plots for pairwise method comparison

We can compare the performance of two methods *A* and *B* by plotting (*A performance*, *B performance*) across numerous data sets

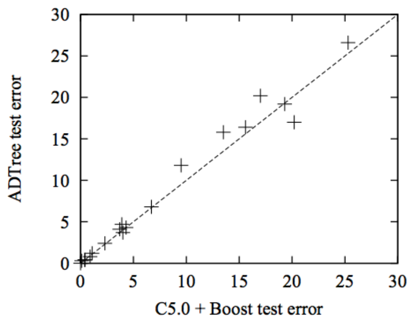


figure from Freund & Mason, *ICML* 1999

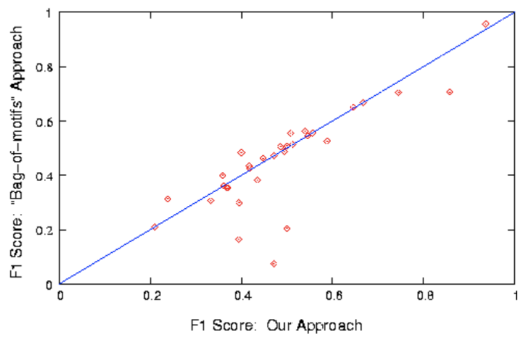


figure from Noto & Craven, *BMC Bioinformatics* 2006

Lesion studies

We can gain insight into what contributes to a learning system's performance by removing (lesioning) components of it

The ROC curves here show how performance is affected when various feature types are removed from the learning representation

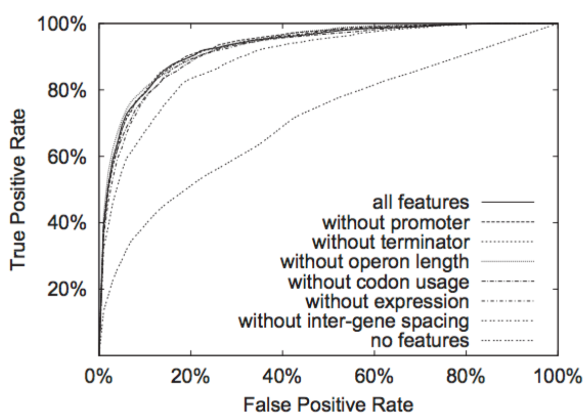


figure from Bockhorst et al., *Bioinformatics* 2003

To avoid pitfalls, ask

1. Is my held-aside test data really representative of going out to collect new data?

    - Even if your methodology is fine, someone may have collected features for positive examples differently than for negatives – should be randomized

    - Example: samples from cancer processed by different people or on different days than samples for normal controls

To avoid pitfalls, ask

2. Did I repeat my entire data processing procedure on every fold of cross-validation, using only the training data for that fold?

    - On each fold of cross-validation, did I ever access in any way the label of a test instance?

    - Any preprocessing done over entire data set (feature selection, parameter tuning, threshold selection) must not use labels

To avoid pitfalls, ask

3. Have I modified my algorithm so many times, or tried so many approaches, on this same data set that I (the human) am overfitting it?

    - Have I continually modified my preprocessing or learning algorithm until I got some improvement on this data set?

    - If so, I really need to get some additional data now to at least test on

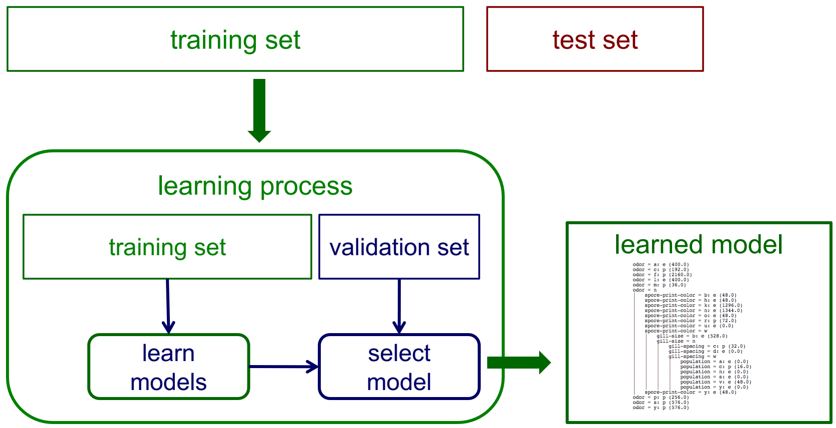
Model selection

- *model selection* is the task of selecting a model from a set of candidate models
    - selecting among decision trees with various levels of pruning
    - selecting *k* in *k*-NN
    - etc.

- one approach to model selection is to use a tuning set or *internal* cross validation

Validation (tuning) sets revisited

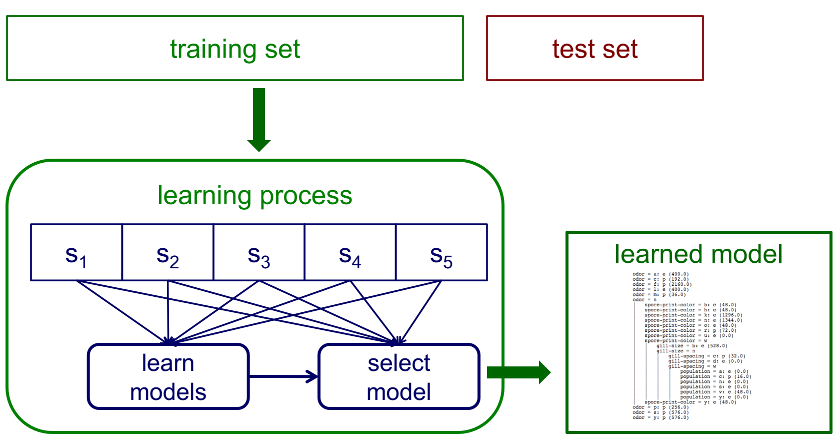
Suppose we want estimates of accuracy during the learning process (e.g. to choose the best level of decision-tree pruning)?



Partition training data into separate training/validation sets

Internal cross validation

Instead of a single validation set, we can use cross-validation within a training set to select a model (e.g. to choose the best level of decision-tree pruning)?



Example: using internal cross validation to select *k* in *k*-NN

given a training set

1. partition training set into n folds, $s_1 \ldots s_n$
2. for each value of k considered
   for i = 1 to n
   learn k-NN model using all folds but s_i
   evaluate accuracy on s_i
3. select k that resulted in best accuracy for $s_1 \ldots s_n$
4. learn model using entire training set and selected k

the steps inside the box are run independently for each training set (i.e. if we're using 10-fold CV to measure the overall accuracy of our *k*-NN approach, then the box would be executed 10 times)