Harvard-MIT Division of Health Sciences and Technology
HST.951J: Medical Decision Support, Fall 2005
Instructors: Professor Lucila Ohno-Machado and Professor Staal Vinterbo

6.873/HST.951 Medical Decision Support

Spring 2005

Cross-validation and Bootstrap Ensembles, Bagging, Boosting

Lucila Ohno-Machado

Training and Tests Sets

- Training set is used to build the model
- Test set left aside for evaluation purposes
- Ideal: different data set to test if model generalizes to other settings
- If data are abundant, then there is no need to “recycle” cases

Cross-validation

- Several training and test set pairs are created
- Results are pooled from all test sets
- “Leave-*n*-out”
- Jackknife (“Leave-1-out”)

Cross-validation
Leave N/2 out

1	23	54	0	1	1	
2	43	23	1	0	1	
3	34	35	0	0	0	→ Training Set — Model Building
4	20	21	1	1	1	
5	19	03	1	1	0	
6	78	04	0	1	0	
7	98	03	0	1	1	
8	35	05	1	1	1	→ Test Set — Evaluation
9	99	23	0	0	1	
10	23	34	0	0	0	

Cross-validation

Leave N/2 out

1 23 54 0 1 1
2 43 23 1 0 1
3 34 35 0 0 0
4 20 21 1 1 1
5 19 03 1 1 0

6 78 04 0 1 0
7 98 03 0 1 1
8 35 05 1 1 1
9 99 23 0 0 1
10 23 34 0 0 0

Training Set | Model Building

Test Set | Evaluation

Leave-N/3-out

1 23 54 0 1 1

2 43 23 1 0 1

3 34 35 0 0 0

4 20 21 1 1 1

5 19 03 1 1 0

6 78 04 0 1 0

7 98 03 0 1 1

8 35 05 1 1 1

9 99 23 0 0 1

10 23 34 0 0 0

Training Set

Model Building

Test Set

Evaluation

Leave-N/3-out

1 23 54 0 1 1
2 43 23 1 0 1
3 34 35 0 0 0

4 20 21 1 1 1
5 19 03 1 1 0
6 78 04 0 1 0

7 98 03 0 1 1
8 35 05 1 1 1
9 99 23 0 0 1
10 23 34 0 0 0

Training Set Model Building

Test Set Evaluation

Leave-N/3-out

1	23	54	0	1	1
2	43	23	1	0	1
3	34	35	0	0	0
4	20	21	1	1	1
5	19	03	1	1	0
6	78	04	0	1	0
7	98	03	0	1	1
8	35	05	1	1	1
9	99	23	0	0	1
10	23	34	0	0	0

Training Set

Model Building

Test Set

Evaluation

Reporting Results

- For each *n*-fold, there will be results from *N*/*n* cases (where *N* is the total number of cases). Collecting all results gives you a test set of *N* previously unseen cases. You can calculate c-index and other statistics from this set.
- Usually, you have to do *k* different randomizations for *n*-fold cross-validation
- Show distribution of indices (e.g., AUC) obtained from different randomization (can also do for different "folds" if they are large enough)
- Show mean and std dev

But what is the final model?

- Several things have been done in practice:
  - Create a model with all cases and report the cross-validation results as a “true” (or at least better than report on the training set performance) estimate of predictive ability
  - Keep an “ensemble” model composed of all models, in which a new case goes to all the models and the result is averaged
    - But some models for some folds are not good at all!
    - Why don’t we ignore or give less weight to the bad models?
      - » See boosting…

Resampling

Bootstrap Motivation

- Sometimes it is not possible to collect many samples from a population
- Sometimes it is not correct to assume a certain distribution for the population
- Goal: Assess sampling variation

Bootstrap

- Efron (Stanford biostats) late 80's
  - "Pulling oneself up by one's bootstraps"
- Nonparametric approach to statistical inference
- Uses *computation* instead of traditional distributional assumptions and asymptotic results
- Can be used for non-linear statistics without known standard error formulas

Example

- Adapted from Fox (1997) “Applied Regression Analysis”
- Goal: Estimate mean difference between Male and Female
- Four pairs of observations are available:

Observ.	Male	Female	Differ. Y
1	24	18	6
2	14	17	-3
3	40	35	5
4	44	41	3

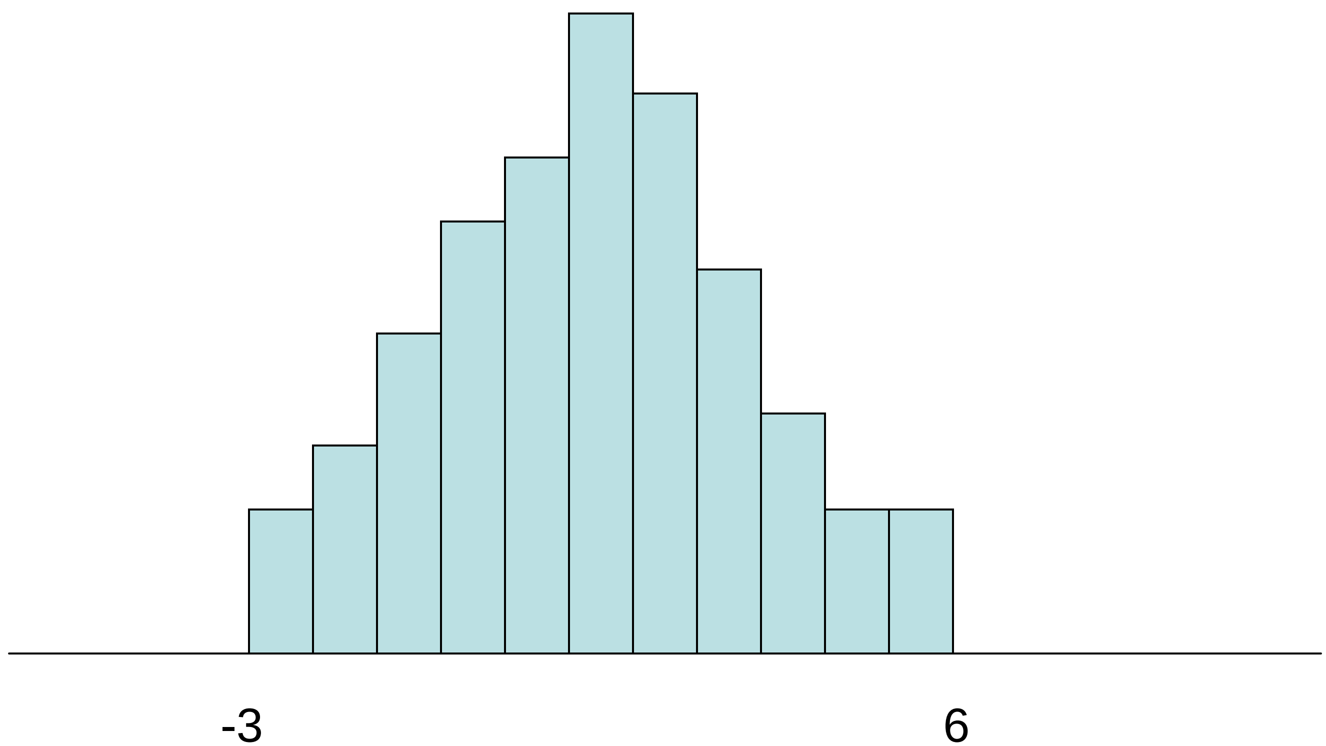
Mean = 2.75

Std Dev = 4.04

Sample with Replacement

Sample	Y_1*	Y_2*	Y_3*	Y_4*	$\bar{Y}$*
1	6	6	6	6	6.00
2	6	6	6	-3	3.75
3	6	6	6	5	5.75
..					
100	-3	5	6	3	2.75
101	-3	5	-3	6	1.25
…					
255	-3	3	3	5	3.5
256	3	3	3	3	3.00

Empirical distribution of Y



The population is to the sample

as

the sample is to the bootstrap samples

In practice (as opposed to previous example), not all bootstrap samples are selected

Procedure

- 1. Specify data-collection scheme that results in observed sample

  Collect(population) -> sample

- 2. Use sample as if it were population (with replacement)

  Collect(sample) -> bootstrap sample1
bootstrap sample 2
etc…

Cont.

- 3. For each bootstrap sample, calculate the estimate you are looking for
- 4. Use the distribution of the bootstrap estimates to estimate the properties of the sample

Bootstrap Confidence Intervals

- Percentile Intervals

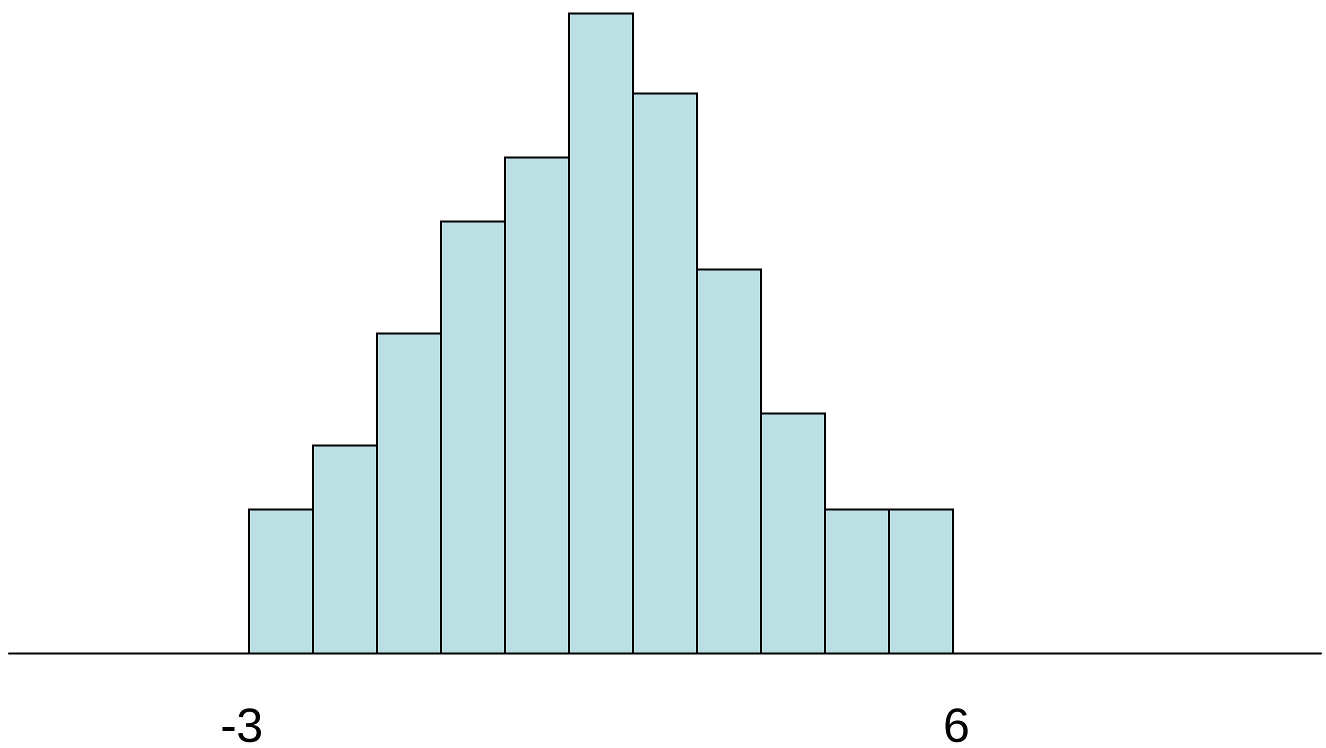
  Example

  – 95% CI is calculated by taking

  – Lower = 0.025 x bootstrap replicates

  – Upper = 0.975 x bootstrap replicates

Empirical distribution of Y



Ensemble Methods: Bagging, Boosting, etc.

Topics

- Bagging
- Boosting
  - Ada-Boosting
  - Arcing
- Stacked Generalization
- Mixture of Experts

Combining classifiers

- Examples: classification trees and neural networks, several neural networks, several classification trees, etc.
- Average results from different models
- Why?
  - Better classification performance than individual classifiers
  - More resilience to noise
- Why not?
  - Time consuming
  - Models become hard to explain

Bagging

- Breiman, 1996
- Derived from bootstrap (Efron, 1993)
- Create classifiers using training sets that are bootstrapped (drawn with replacement)
- Average results for each case

Bagging Example (Opitz, 1999)

Original	1	2	3	4	5	6	7	8
Training set 1	2	7	8	3	7	6	3	1

Bagging Example (Opitz, 1999)

Original	1	2	3	4	5	6	7	8
Training set 1	2	7	8	3	7	6	3	1
Training set 2	7	8	5	6	4	2	7	1

Bagging Example (Opitz, 1999)

Original	1	2	3	4	5	6	7	8
Training set 1	2	7	8	3	7	6	3	1
Training set 2	7	8	5	6	4	2	7	1
Training set 3	3	6	2	7	5	6	2	2

Bagging Example (Opitz, 1999)

Original	1	2	3	4	5	6	7	8
Training set 1	2	7	8	3	7	6	3	1
Training set 2	7	8	5	6	4	2	7	1
Training set 3	3	6	2	7	5	6	2	2
Training set 4	4	5	1	4	6	4	3	8

Boosting

- A family of methods
- Sequential production of classifiers
- Each classifier is dependent on the previous one, and focuses on the previous one’s errors
- Examples that are incorrectly predicted in previous classifiers are chosen more often or weighted more heavily

Boosting Example (Opitz, 1999)

Original	1	2	3	4	5	6	7	8
Training set 1	2	7	8	3	7	6	3	1

Boosting Example (Opitz, 1999)

Original	1	2	3	4	5	6	7	8
Training set 1	2	7	8	3	7	6	3	1
Training set 2	1	4	5	4	1	5	6	4

Boosting Example (Opitz, 1999)

Original	1	2	3	4	5	6	7	8
Training set 1	2	7	8	3	7	6	3	1
Training set 2	1	4	5	4	1	5	6	4
Training set 3	7	1	5	8	1	8	1	4

Boosting Example (Opitz, 1999)

Original	1	2	3	4	5	6	7	8
Training set 1	2	7	8	3	7	6	3	1
Training set 2	1	4	5	4	1	5	6	4
Training set 3	7	1	5	8	1	8	1	4
Training set 4	1	1	6	1	1	3	1	5

Ada-Boosting

- Freund and Schapire, 1996
- Two approaches
  - Select examples according to error in previous classifier (more representatives of misclassified cases are selected) – more common
  - Weigh errors of the misclassified cases higher (all cases are incorporated, but weights are different) – not for all algorithms

Ada-Boosting

- Define ε_k as the sum of the probabilities for the misclassified instances for current classifier C_k
- Multiply probability of selecting misclassified cases by

$$\beta_k = (1 - \varepsilon_k) / \varepsilon_k$$

- "Renormalize" probabilities (i.e., rescale so that it sums to 1)
- Combine classifiers $C_1 \ldots C_k$ using weighted voting where C_k has weight $\log(\beta_k)$

Arcing

- Arcing-x4 (Breiman, 1996)
- For the *i*th example in the training set, m_i refers to the number of times that it was misclassified by the previous *K* classifiers
- Probability p_i of selecting example *i* in the next classifier is

$$p_i = \frac{1+m_i^4}{\sum_{j=1}^{N} 1+m_j^4}$$

- Empirical determination

Empirical comparison (Opitz, 1999)

- 23 data sets from UCI repository
- 10-fold cross validation
- Backpropagation neural nets
- Classification trees
- Simple (multiple NNs with different initial weights), Bagging, Ada-boost, Arcing
- Correlation coefficients of estimates from different ensembles

Opitz, D. and Maclin, R. (1999) "Popular Ensemble Methods: An Empirical Study", Journal of Artificial Intelligence Research, Volume 11, pages 169-198.

Correlation coefficients

	Neural Net				Classification Tree		
	Simple	Bagging	Arcing	Ada	Bagging	Arcing	Ada
Simple NN	1	.88	.87	.85	-.10	.38	.37
Bagging NN	.88	1	.78	.78	-.11	.35	.35
Arcing NN	.87	.78	1	.99	.14	.61	.60
Ada NN	.85	.78	.99	1	.17	.62	.63
Bagging CT					1	.68	.69
Arcing CT					.68	1	.96
Ada CT					.69	.96	1

Results

- Ensembles generally better than single, but not so different from “Simple” (NNs with different initial random weights)
- Ensembles within NNs and CTs are strongly correlated
- Ada-boosting and arcing strongly correlated even across different algorithms (boosting may depend more on data set than type of classifier algorithm)
- 40 networks in ensemble were sufficient
- NNs generally better than CTs

More results

- Created data sets with different levels of noise (random selection of possible value for a feature or outcome) from the 23 sets
- Created artificial data with noise

Conclusion:

- Boosting worse with more noise

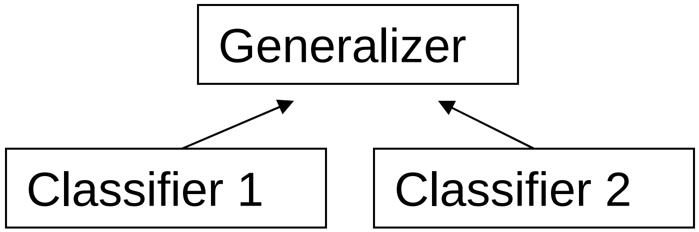
Other work

- Opitz and Shavlik
  - Genetic search for classifiers that are accurate yet different
- Create diverse classifiers by:
  - Using different parameters
  - Using different training sets

Opitz, D. & Shavlik, J. (1999). A Genetic Algorithm Approach for Creating Neural Network Ensembles. *Combining Artificial Neural Nets.* Amanda Sharkey (ed.). (pp. 79-97). Springer-Verlag, London.

Stacked Generalization

- Wolpert, 1992
- Level-0 models are based on different learning models and use original data (level-0 data)
- Level-1 models are based on results of level-0 models (level-1 data are outputs of level-0 models) -- also called “generalizer”



Empirical comparison

- Ting, 1999
- Compare SG to best model and to arcing and bagging
- Stacked C4.5, naïve Bayes, and a nearest neighbor learner
- Used multi-response linear regression as generalizer

Ting, K.M. & Witten, I.H., *Issues in Stacked Generalization*. Journal of Artificial Intelligence Research. AI Access Foundation and Morgan Kaufmann Publishers, Vol.10, pp. 271-289, 1999.

Results

- SG had better performance (accuracy) than best level-0 model
- Use of continuous estimates better than use of predicted class
- Better than majority vote
- Similar performance as arcing and bagging
- Good for parallel computation (like bagging)

Related work

- Decomposition of problem into subtasks
- Mixture of experts (Jacobs, 1991)
  - Each expert here takes care of a certain input space
- Hierarchical neural networks
  - Cases are routed to pre-defined expert networks

Jacobs, R. A., Jordan, M. I., Nowlan, S. J., & Hinton, G. E. (1991) *Adaptive mixtures of local experts*. In Neural Computation 3, pp. 79-87, MIT press.

Ideas for final projects

- Compare single, bagging, and boosting on other classifiers (e.g., logistic regression, rough sets)
- Reproduce previous comparisons using different data sets
- Use other performance measures
- Study the effect of voting scheme
- Try to find a relationship between initial performance, number of cases, and number of classifiers within an ensemble
- Genetic search for good diverse classifiers
- Analyze effect of prior outlier removal on boosting

Variable Selection

- Ideal: consider all variable combinations
  - Not feasible in most data sets with large number of n variables: 2^n
- Greedy Forward:
  - Select most important variable as the “first component”, Select other variables conditioned on the previous ones
  - Stepwise: consider backtracking
- Greedy Backward:
  - Start with all variables and remove one at a time.
  - Stepwise: consider backtracking
- Other search methods: genetic algorithms that optimize classification performance and # variables

Variable Selection

- Use few variables (genes)
- Interpretation is easier
- Cheaper
- More cases can be used (fewer missing values)