BIG DATA ANALYTICS
Resampling methods

Olga Klopp
klopp@essec.edu

ESSEC BUSINESS SCHOOL
CentraleSupélec

Resampling methods

- *Resampling methods involve repeatedly drawing samples from a training set and refitting a model of interest on each sample in order to obtain additional information about the model*
- E.g., to estimate the variability of a linear regression fit:
  - we can repeatedly draw different samples from the training data
  - fit a linear regression to each new sample
  - examine the extent to which the resulting fits differ
- It allows to obtain information that would not be available from fitting the model only once using the original training sample.

Resampling methods

- Can be computationally expensive: involves fitting the same statistical method multiple times
- Two of the most commonly used resampling methods: **cross-validation and the bootstrap**
- Both are important tools in the practical application of many statistical learning procedures:
  - E.g., cross-validation can be used to estimate the test error or to select the appropriate level of flexibility
- The process of evaluating a models performance: *model assessment*
- The process of selecting the proper level of flexibility: *model selection*

Outline

Outline

Test and Training errors

- The *test error* is the average error that results from using a statistical learning method to predict the response on a new observation
- **The use of a particular statistical learning method is justified if it results in a low test error**
- The *training error* is calculated by applying the statistical learning method to the observations used in its training
- The training error rate often is quite different from the test error rate: the former can dramatically underestimate the latter

Test and Training errors

- A number of techniques can be used to estimate the test error using the available training data
- A class of methods that estimate the test error rate by holding out a subset of the training observations from the fitting process: *cross-validation*

The Validation Set Approach

- The validation set approach:
  - Divides randomly the available set of observations into two parts: a training set and a validation set (or hold-out set)
  - The model is fit on the training set
  - The fitted model is used to predict the responses for the observations in the validation set
  - The resulting validation set error rate provides an estimate of the test error rate

Example

- Example: assume that performing linear regression analysis we found out that there is a non-linear relationship between the input and output variables
- So we may ask if a quadratic or higher-order fit might provide better results
- We can answer this question by looking at the p-values associated with a quadratic term and higher-order polynomial terms in a linear regression
- We can also answer this question using the validation method

Example

- We randomly split the observations: a training set and a validation set
- We fit various regression models on the training sample (e.g. with quadratic term, with cubic term and etc)
- Compare the validation set error rates that result from fitting
- Pick the model with smaller validation set error.

Drawbacks

- Simple and is easy to implement
- Two potential drawbacks:
  1. The validation estimate of the test error rate can be highly variable
  2. Only a subset of the observations are used to fit the model:
     - Statistical methods tend to perform worse when trained on fewer observations
     - The validation set error rate may overestimate the test error rate for the model fit on the entire data set

Cross-validation, a refinement of the validation set approach can address these two issues

Leave-One-Out Cross-Validation

- Leave-one-out cross-validation (LOOCV) also involves splitting the set of observations into two parts
- A single observation (x_1, y_1) is used for the validation set, and the remaining observations $\{(x_2, y_2), \ldots, (x_n, y_n)\}$ make up the training set
- The statistical learning method is fit on the $n-1$ training observations, and a prediction $\hat{y}_1$ is made for the excluded observation, using its value x_1
- $MSE_1 = (y_1 - \hat{y}_1)^2$ provides an approximately unbiased estimate for the test error
- MSE_1 is a poor estimate because as it is based upon a single observation (x_1, y_1)

Leave-One-Out Cross-Validation

- We can repeat the procedure by selecting (x_2, y_2) for the validation data and computing $MSE_2 = (y_2 - \hat{y}_2)^2$
- Repeating this approach n times produces n squared errors: $MSE_1, \ldots, MSE_n$
- The LOOCV estimate for the test MSE is the average of these n test error estimates:

$$CV_{(n)} = \frac{1}{n} \sum_{i=1}^{n} MSE_i$$

Leave-One-Out Cross-Validation

- Major advantages over the validation set approach:

  1. It has far less bias:
     - we repeatedly fit the statistical learning method using training sets that contain $n-1$ observations, almost as many as are in the entire data set
     - the LOOCV approach tends not to overestimate the test error rate
  2. Performing LOOCV multiple times will always yield the same results: there is no randomness in the training/validation set splits.

Leave-One-Out Cross-Validation

- LOOCV may be expensive to implement, since the model has to be fitted n times
- Can be very time consuming if n is large, and if each individual model is slow to fit
- With least squares linear or polynomial regression: the cost of LOOCV the same as that of a single model fit

Leave-One-Out Cross-Validation

- $CV_{(n)}$ can be computed after estimating the model once on the complete data set:

$$CV_{(n)} = \frac{1}{n} \sum_{i=1}^{n} \left(\frac{y_i - \hat{y}_i}{1 - h_i} \right)^2$$

  $h_i = \frac{1}{n} + \frac{(x_i - \bar{x})^2}{\sum_{k=1}^{n} (x_k - \bar{x})^2}$ is the leverage and $\hat{y}_i$ is the ith fitted value from the original least squares fit

- This is like the ordinary MSE, except the ith residual is divided by $1 - h_i$

- The leverage lies between $1/n$ and 1, and reflects the amount that an observation influences its own fit

Leave-One-Out Cross-Validation

- LOOCV is a very general method, and can be used with any kind of predictive modeling: logistic regression, linear discriminant analysis...
- The magic formula does not hold in general, in which case the model has to be refit n times

k-Fold Cross-Validation

- k-fold CV: randomly divide the set of observations into k groups, or folds, of approximately equal size
- The first fold is treated as a validation set, and the method is fit on the remaining $k-1$ folds
- The mean squared error, MSE_1, is computed on the observations in the held-out fold

k-Fold Cross-Validation

- This procedure is repeated k times
- k estimates of the test error

$$MSE_1, MSE_2, \ldots, MSE_k$$

- The k-fold CV estimate is computed by averaging these values:

$$CV_k = \frac{1}{k}\sum_{i=1}^{k} MSE_i$$

k-Fold Cross-Validation

- LOOCV is a special case of k-fold CV with $k = n$
- In practice, one performs k-fold CV using $k = 5$ or $k = 10$
- Computational advantage: LOOCV requires fitting the statistical learning method n times
- This can be computationally expensive
- In contrast, performing 10-fold CV requires fitting the learning procedure only ten times, which may be much more feasible

Bias-Variance Trade-Off for k-Fold Cross-Validation

An important advantage of k-fold CV is that it often gives more accurate estimates of the test error rate than does LOOCV!

- The validation set approach overestimates the test error rate as the training set contains only half of the observations
- LOOCV will give approximately unbiased estimates of the test error, since each training set contains $n - 1$ observations
- Performing k-fold CV for, $k = 5$ or $k = 10$ will lead to an intermediate level of bias: each training set contains $(k - 1)n/k$ observations

Bias-Variance Trade-Off for k-Fold Cross-Validation

- From the perspective of bias reduction, it is clear that LOOCV is to be preferred to k-fold CV
- The bias is not the only source for concern in an estimating procedure: we must also consider the procedures variance!
- LOOCV has higher variance than does k-fold CV with $k < n$

Bias-Variance Trade-Off for k-Fold Cross-Validation

- When we perform LOOCV:
  - we are averaging the outputs of n fitted models
  - each of which is trained on an almost identical set of observations
  - these outputs are highly correlated with each other
- When we perform k-fold CV with $k < n$:
  - we are averaging the outputs of k fitted models
  - they are less correlated with each other, since the overlap between the training sets in each model is smaller

Bias-Variance Trade-Off for k-Fold Cross-Validation

- The mean of many highly correlated quantities has higher variance
- The test error estimate resulting from LOOCV tends to have higher variance than the test error estimate resulting from k-fold CV
- **There is a bias-variance trade-off associated with the choice of k in k-fold cross-validation**

Model Selection

- If the goal is to determine how well a given statistical learning procedure performs on independent data $\implies$ the actual estimate of the test MSE
- If the goal is to identify the method that results in the lowest test error $\implies$ it is enough to look at the location of the minimum point in the estimated test MSE curve

Cross-Validation on Classification Problems

- When we use the cross-validation in the regression setting where the outcome Y is quantitative we may use MSE to quantify test error
- Cross-validation can also be a very useful approach in the classification setting when Y is qualitative
- In this setting, cross-validation works exactly in the same way: instead of MSE we use the number of misclassified observations

Outline

The Bootstrap

- Widely applicable and powerful statistical tool to quantify the uncertainty associated with a given estimator or statistical learning method
- **Bootstrap can be easily applied to a wide range of statistical learning methods, including some for which a measure of variability is otherwise difficult to obtain**

Example

- Determine the best investment allocation (simulated data)
- We wish to invest a fixed sum of money in two financial assets that yield returns of X and Y
  - X and Y are random quantities
- We will invest a fraction α of our money in X, and will invest the remaining $1 - \alpha$ in Y
- Choose α to minimize the total risk, or variance, of our investment
- We want to minimize

$$\mathrm{Var}(\alpha X + (1 - \alpha) Y)$$

Example

- The value that minimizes the risk:

$$\alpha = \frac{\sigma_Y^2 - \sigma_{XY}}{\sigma_X^2 + \sigma_Y^2 - 2\sigma_{XY}}$$

- σ_X^2, σ_Y^2 and σ_{XY} are unknown
- Compute estimates for these quantities using a data set that contains past measurements for X and Y
- Compute estimate of the value of α that minimizes the variance of our investment:

$$\hat{\alpha} = \frac{\hat{\sigma}_Y^2 - \hat{\sigma}_{XY}}{\hat{\sigma}_X^2 + \hat{\sigma}_Y^2 - 2\hat{\sigma}_{XY}}$$

Example

- Accuracy of our estimate of α?
- As we simulated the data, we know the exact values for $\sigma_X^2 = 1$, $\sigma_Y^2 = 1.25$, and $\sigma_{XY} = 0.5 \implies$ we know the true value: $\alpha = 0.6$
- We repeated the process of simulating observations of X and Y, and estimating α 1,000 times:

$$\hat{\alpha}_1, \hat{\alpha}_2, \ldots, \hat{\alpha}_{1000}$$

- The mean over all 1,000 estimates:

$$\frac{1}{1000} \sum_{i=1}^{1000} \hat{\alpha}_i = 0.5996$$

  very close to $\alpha = 0.6$!
- The standard deviation of the estimates is 0.083

Bootstrap

- This gives us a very good idea of the accuracy of $\hat{\alpha}$: for a random sample from the population, we would expect $\hat{\alpha}$ to differ from α by approximately 0.08
- In practice this procedure can not be applied: for real data we can not generate new samples from the original population

Bootstrap

- *The bootstrap approach allows us to use a computer to emulate the process of obtaining new sample sets*
- We can estimate the variability of the estimator without generating additional samples
- Rather than repeatedly obtaining independent data sets from the population, we instead obtain distinct data sets by repeatedly sampling observations from the original data set

Bootstrap

- Assume that we have a data set Z of size n
- We randomly select n observations from the data set in order to produce a bootstrap data set: Z_1
- The sampling is performed **with replacement**
  - the same observation can occur more than once in the bootstrap data set
- This procedure is repeated B times for some large value of B

Bootstrap

- We have B different bootstrap data sets:

$$Z_1, Z_2, \ldots, Z_B$$

- We produce B corresponding estimates: $\hat{\alpha}_1, \hat{\alpha}_2, \ldots, \hat{\alpha}_B$
- The estimate of standard error of $\hat{\alpha}$ (estimated from the original data set):

$$\mathrm{SE}_B(\hat{\alpha}) = \sqrt{\frac{1}{B-1} \sum_{r=1}^{B} \left(\hat{\alpha}_r - \frac{1}{B} \sum_{i=1}^{B} \hat{\alpha}_i \right)^2}$$

For our example, the bootstrap estimate $\mathrm{SE}_B(\hat{\alpha})$ is 0.087, very close to the estimate of 0.083 obtained using 1,000 simulated data sets!

Outline

Bootstrap on the large data sets?

- The bootstrap provides a simple and powerful mean of assessing the quality of estimators
- Requires repeated estimator computation on resamples having size comparable to that of the original dataset
- If the original dataset is large, then this computation can be costly

In settings involving large datasets, the computation of bootstrap-based quantities can be prohibitively demanding

Bootstrap on the large data sets?

- As the amount of available data grows, the number of parameters to be estimated and the number of potential sources of bias often also grow
- Requires to be able to tractably assess estimator quality in the setting of large data
- An automatic, accurate mean of assessing estimator quality that is scalable to large datasets?

Bag of Little Bootstraps

- An interesting alternative is the **Bag of Little Bootstraps (BLB)** **[Kleiner et al, 2012]**
  - Incorporates features of both the bootstrap and subsampling
  - Provides a robust, computationally efficient mean of assessing estimator quality
  - Is well suited to modern parallel and distributed computing architectures
  - Retains the generic applicability, statistical efficiency, and favorable theoretical properties of the bootstrap

Bag of Little Bootstraps

Idea: Bag of Little Bootstraps functions by combining the results of bootstrapping multiple small subsets of a larger original dataset

- We observe a sample $X_1, ..., X_n$ and based on this sample we obtain an estimate $\hat{\theta}_n$
  - $\hat{\theta}_n$ might estimate a measure of correlation, the parameters in a linear regression, or the prediction accuracy of a trained classification model
- Let ξ be the estimator of quality assessment
  - e.g. a confidence region, a standard error, or a bias

Bag of Little Bootstraps

- Given a subset size $b < n$, BLB samples s subsets of size b from the original n data points, uniformly at random
- BLB estimates error ξ using $\frac{1}{s}\sum_{i=1}^{s}\xi_i$
- Each ξ_i is computed in the manner of the bootstrap: we repeatedly resample n points i.i.d. and compute the estimate on each resample

Computational benefits

- The substantial computational benefits: each BLB resample, despite having nominal size n has at most b different values
- We can represent each resample by maintaining at most b distinct points, accompanied by corresponding sampled counts $\implies$ each resample requires storage space in $O(b)$

Computational benefits

- If the estimator can work directly with this weighted data representation, then its computational requirements - with respect to both time and storage space - scale only in b, rather than n
  - This property does holds for many if not most commonly used estimators

BLB only requires repeated computation on small subsets of the original dataset

Computational benefits

- Each bootstrap resample contains approximately $0.632n$ distinct points, which is large if n is large
- In contrast, each BLB resample contains at most b distinct points, and b can be chosen to be much smaller than n or $0.632n$

Computational benefits

- For example, we might take $b = n^\gamma$ where $\gamma \in [0.5, 1]$
- If $n = 1,000,000$ then each bootstrap resample would contain approximately $632,000$ distinct points
- With $b = n^{0.6}$ each BLB subsample and resample would contain at most $3,981$ distinct points
- If each data point occupies 1 MB of storage space, then the original dataset would occupy 1 TB
- A bootstrap resample would occupy approximately 632 GB
- Each BLB subsample or resample would occupy at most 4 GB.

Computational benefits

BLB has a significantly more favorable computational profile than the bootstrap and reaches comparably high accuracy

References

- Trevor Hastie, Robert Tibshirani, Jerome Friedman. *The elements of statistical learning - Data Mining, inference, and prediction.* Springer.
- Gareth James, Daniela Witten, Trevor Hastie, Robert Tibshirani. *An Introduction to Statistical Learning with applications in R.* Springer.
- Kleiner, A., Talwalkar, A., Sarkar, P., and Jordan, M. I. A scalable bootstrap for massive data. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* Volume 76, Issue 4 (2014)