CHAPTER 4
OVER-FITTING AND MODEL TUNING

APPLIED PREDICTIVE MODELING BY KUHN & JOHNSON
COLLATED BY PROF. CHING-SHIH (VINCE) TSOU (PH.D.)
CENTER FOR APPLICATIONS OF DATA SCIENCE (CADS)
GRADUATE INSTITUTE OF INFORMATION AND DECISION SCIENCES (GIIDS)
NATIONAL TAIPEI UNIVERSITY OF BUSINESS (NTUB)
CHINESE ACADEMY OF R SOFTWARE (CARS)
DATA SCIENCE & BUSINESS APPLICATIONS (DSBA) ASSOCIATION OF TAIWAN

AGENDA

- Introduction
- The Problem of Over-Fitting
- Model Tuning
- Data Splitting
- Resampling Techniques
- Case Study: Credit Scoring
- Choosing Final Tuning Parameters
- Data Splitting Recommendations
- Choosing Between Models
- Computing

INTRODUCTION – IMPORTANCE OF OVERFITTING ISSUE

- Over-fitting has been discussed in the fields of:
  - Forecasting (Clark 2004).
  - Medical Research (Simon et al. 2003 ; Steyerberg 2010).
  - Chemometrics (Gowen et al. 2010 ; Hawkins 2004 ; Defernez and Kemsley 1997).
  - Meteorology (Hsieh and Tang 1998).
  - Finance (Dwyer 2005).
  - Marital Research (Heyman and Slep 2001).
- These references illustrate that over-fitting is a concern for any predictive model regardless of field of research.

INTRODUCTION

- The aim of this chapter is to explain and illustrate key principles of laying a foundation onto which trustworthy models can be built and subsequently used for prediction.
- We will describe strategies that enable us to have confidence that the model we build will predict new samples with a similar degree of accuracy on the set of data for which the model was evaluated (i.e. validation or test set).
- Without this confidence, the model's predictions are *useless*.

INTRODUCTION – DATA QUALITY AND REPRESENTATIVE

- For many problems, the data may have a limited number of samples, may be of less-than-desirable quality, and/or may be unrepresentative of future samples.
- While there are ways to build predictive models on small data sets, which we will describe in this chapter, we will assume that data quality is sufficient and that it is representative of the entire sample population. (Real-world data, even though not *small,* probably do not conform to this assumption.)
- Working under these assumptions, we must use the data at hand to find the best predictive model.

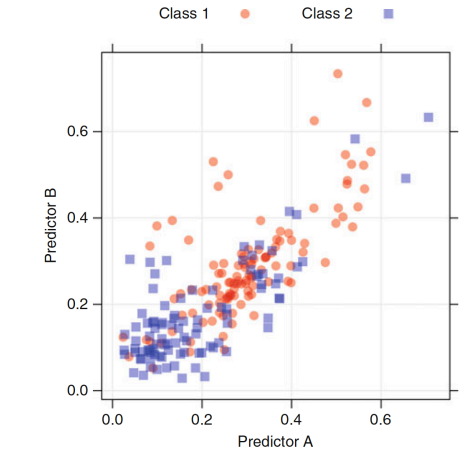
INTRODUCTION – MODEL TUNING AND EVALUATION PROCESS

- Traditionally, this has been achieved by splitting the existing data into training and test sets.
- The training set is used to (1) build and tune the model and the test set is used to (2) estimate the model's predictive performance.
- Modern approaches to model building split the data into multiple training and testing sets, which have been shown to often find more optimal tuning parameters and give a more accurate representation of the model's predictive performance.
- To *avoid over-fitting*, we propose a general model building approach that encompasses model tuning and model evaluation with the ultimate goal of finding the *reproducible structure* in the data.

THE PROBLEM OF OVER-FITTING – FROM NOISE TO SIGNAL

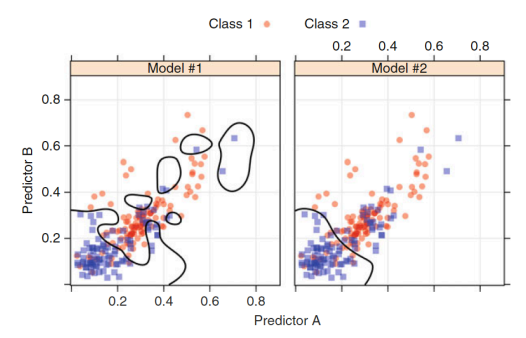
- In addition to learning the *general patterns* in the data, the model has also learned the characteristics of each sample's *unique noise*. (general pattern vs. unique noise)
- This type of model is said to be over-fit and will usually have *poor* accuracy when predicting a *new* sample.
- To illustrate over-fitting and other concepts in this chapter, consider the simple classification example in next slide's figure.

THE PROBLEM OF OVER-FITTING – A SIMPLE CLASSIFICATION EXAMPLE



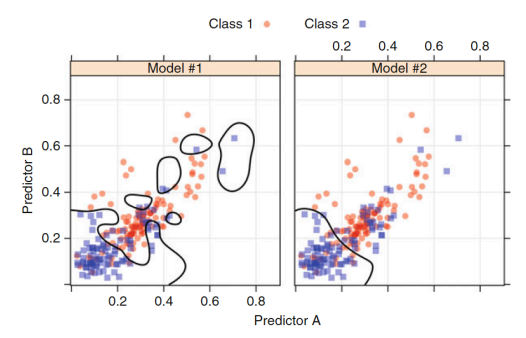
- These data contain 208 samples that are designated either as "Class 1" or "Class 2."
- The classes are *fairly balanced*; there are 111 samples in the first class and 97 in the second.
- Furthermore, there is a *significant overlap* between the classes which is often the case for most applied modeling problems. (Hard to separate them.)

THE PROBLEM OF OVER-FITTING



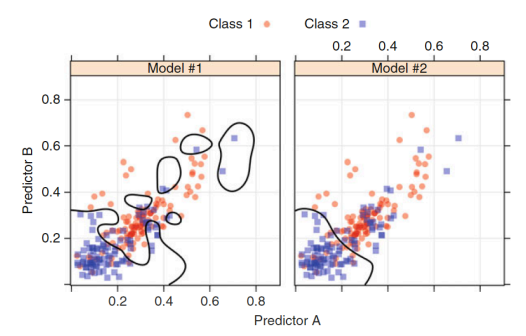
- One objective for a data set such as this would be to develop a model to classify new samples.
- In this two-dimensional example, the classification models or rules can be represented by boundary lines.
- The lines envelop the area where each model predicts the data to be the second class (blue squares).

THE PROBLEM OF OVER-FITTING



- The left-hand panel ("Model #1") shows a boundary that is complex and attempts to encircle every possible data point.
- The right-hand panel ("Model #2") shows an alternative model fit where the boundary is fairly smooth and does not overextend itself to correctly classify every data point in the training set.
- The pattern in this panel is not likely to generalize to new data.

THE PROBLEM OF OVER-FITTING



- To gauge how well the model is classifying samples, one might use the training set.
- In doing so, the estimated error rate for the model in the left-hand panel would be overly optimistic. (re-substitution error rate usually optimistic)

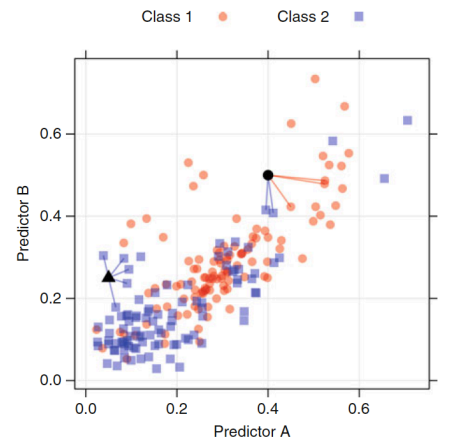
THE PROBLEM OF OVER-FITTING

- Estimating the utility of a model by re-predicting the training set is referred to apparent performance of the model (e.g., the apparent error rate).
- In two dimensions, it is not difficult to visualize that one model is over-fitting, but most modeling problems are in much higher dimensions. (but more difficult to visualize model in high dimensional space)
- In these situations, it is very important to have a tool for characterizing how much a model is over-fitting the training data.

MODEL TUNING

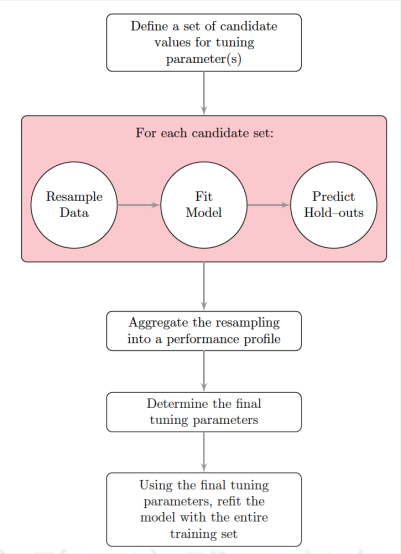
- Many models have important parameters which cannot be directly estimated from the data.
- For example, in the K-nearest neighbor classification model, a new sample is predicted based on the K-closest data points in the training set.
- The question remains as to how many neighbors should be used.
- A choice of too few neighbors may over-fit the individual points of the training set while too many neighbors may not be sensitive enough to yield reasonable performance.
- This type of model parameter is referred to as a tuning parameter because there is no analytical formula available to calculate an appropriate value.

MODEL TUNING



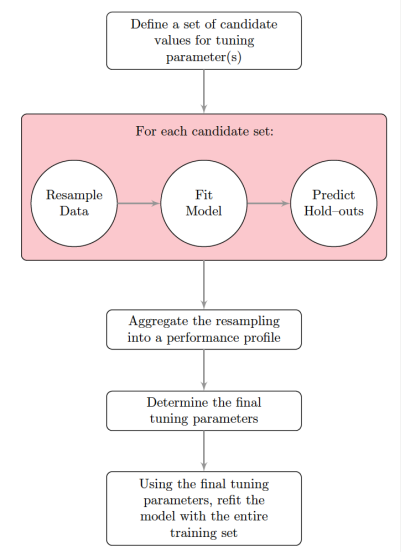
- Two new samples are being predicted.
- One sample (●) is near a mixture of the two classes; three of the five neighbors indicate that the sample should be predicted as the first class.
- The other sample (▲) has all five points indicating the second class should be predicted.

MODEL TUNING – PARAMETER TUNING PROCESS



- Once a candidate set of parameter values has been selected, then we must obtain trustworthy estimates of model performance.
- The performance on the hold-out samples is then aggregated into a performance profile which is then used to determine the final tuning parameters.
- We then build a final model with all of the training data using the selected tuning parameters.

MODEL TUNING – PARAMETER TUNING PROCESS



- The training data would then be resampled and evaluated many times for each tuning parameter value.
- These results would then be aggregated to find the optimal value of K.
- The procedure defined in this figure uses a set of candidate models that are defined by the tuning parameters.

MODEL TUNING – PERFORMANCE EVALUATION AFTER BUILDING

- A more difficult problem is obtaining trustworthy estimates of model performance for these candidate models.
- As previously discussed, the apparent error rate (calculated by training set) can produce extremely optimistic performance estimates.
- A better approach is to test the model on samples that were not used for training.
- Evaluating the model on a test set is the obvious choice, but, to get reasonable precision of the performance values, the size of the test set may need to be large.

DATA SPLITTING – THE HEART OF THE PROCESS

- Now that we have outlined the general procedure for finding optimal tuning parameters, we turn to discussing the heart of the process: data splitting.
- A few of the common steps in model building are:
  - Pre-processing the predictor data
  - Estimating model parameters
  - Selecting predictors for the model
  - Evaluating model performance
  - Fine tuning class prediction rules (via ROC curves, etc.)

DATA SPLITTING

- When a large amount of data is at hand, a set of samples can be set aside to evaluate the final model.
- The "training" data set is the general term for the samples used to *create* the model, while the "test" or "validation" data set is used to *qualify* performance.
- However, when the number of samples is *not large*, a strong case can be made that a test set should be avoided because *every sample may be needed for model building*.
- Additionally, the size of the test set may not have sufficient power or precision to make reasonable judgements.

DATA SPLITTING – NONRANDOM APPROACH

- If a test set is deemed necessary, there are several methods for splitting the samples.
- Nonrandom approaches to splitting the data are sometimes appropriate. For example,
  - If a model was being used to predict patient outcomes, the model may be created using certain patient sets (e.g., from the same clinical site or disease stage), and then tested on a different sample population to understand how well the model generalizes.
  - In chemical modeling for drug discovery, new "chemical space" is constantly being explored. We are most interested in accurate predictions in the chemical space that is currently being investigated rather than the space that was evaluated years prior. The same could be said for spam filtering; it is more important for the model to catch the new spamming techniques rather than prior spamming schemes.

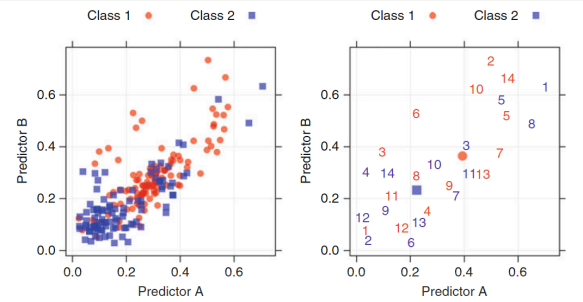
DATA SPLITTING – RANDOM APPROACH

- The simplest way to split the data into a training and test set is to take a simple random sample.
- This does not control for any of the data attributes, such as the percentage of data in the classes.
- When one class has a disproportionately small frequency compared to the others, there is a chance that the distribution of the outcomes may be substantially different between the training and test sets.
- When the outcome is a number, a similar strategy can be used; the numeric values are broken into similar groups (e.g., low, medium, and high) and the randomization is executed within these groups.

DATA SPLITTING – DISSIMILARITY CONSIDERATION

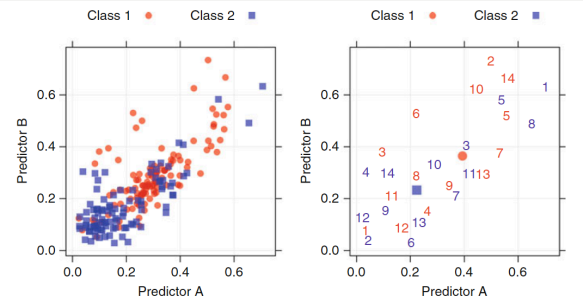
- Dissimilarity between two samples can be measured in a number of ways.
- The simplest method is to use the distance between the predictor values for two samples.
- If the distance is small, the points are in close proximity. Larger distances between points are indicative of dissimilarity.
- To use dissimilarity as a tool for data splitting, suppose the test set is initialized with a single sample.
- The dissimilarity between this initial sample and the unallocated samples can be calculated.

DATA SPLITTING



- Dissimilarity sampling was conducted separately within each class. (attentive to the class distribution)
- First, a sample within each class was chosen to start the process (designated as ■ and ● in the figure).
- For the first class, the most dissimilar point was in the extreme Southwest of the initial sample.
- On the second round, the dissimilarities were aggregated using the minimum (as opposed to the average).

DATA SPLITTING



- Again, for the first class, the chosen point was far in the Northeast of the predictor space.
- As the sampling proceeds, samples were selected on the periphery of the data then work inward. (samples are selected peripherally then moving inward)

RESAMPLING TECHNIQUES

RESAMPLING TECHNIQUES

- One of the tuning parameters for this model sets the price for misclassified samples in the training set and is generally referred to as the "cost" parameter.
- When the cost is large, the model will go to great lengths to correctly label every point (as in the left panel) while smaller values produce models that are not as aggressive.
- The class boundary in the left panel was created by manually setting the cost parameter to a very high number.

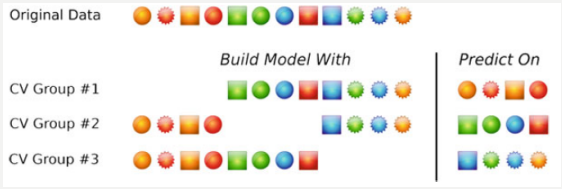
RESAMPLING TECHNIQUES

- What is resampling?
  - Generally, resampling techniques for estimating model performance operate similarly: a subset of samples are used to fit a model and the remaining samples are used to estimate the efficacy of the model.
  - This process is repeated multiple times and the results (performance) are aggregated and summarized.
  - The differences in techniques usually center around the method in which subsamples are chosen.
- We will consider the main flavors of resampling in the next few subsections.
  - k-Fold Cross-Validation
  - Generalized Cross-Validation
  - Repeated Training/Test Splits
  - The Bootstrap

K-FOLD CROSS-VALIDATION

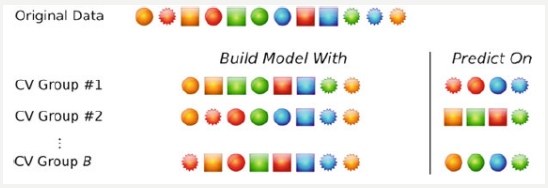
- The samples are randomly partitioned into k sets of roughly equal size.
- A model is fit using the all samples except the first subset (called the first fold).
- The first subset is returned to the training set and procedure repeats with the second subset held out, and so on.
- From a practical viewpoint, larger values of k are more computationally burdensome.
- In the extreme, LOOCV is most computationally taxing because it requires as many model fits as data points and each model fit uses a subset that is nearly the same size of the training set.
- Research (Molinaro 2005 ; Kim 2009) indicates that repeating k-fold cross-validation can be used to effectively increase the precision of the estimates while still maintaining a small bias.

K-FOLD CROSS-VALIDATION



- Twelve training set samples are represented as symbols and are allocated to three groups.
- These groups are left out in turn as models are fit.
- The average of the three performance estimates would be the cross-validation estimate of model performance.
- In practice, the number of samples in the held-out subsets can vary but are roughly equal size.

REPEATED TRAINING/TEST SPLITS



- Repeated training/test splits is also known as "leave-group-out cross-validation" or "Monte Carlo cross-validation."
- This technique simply creates multiple splits of the data into modeling and prediction sets.
- One difference between this procedure and k -fold cross-validation are that samples can be represented in multiple held-out subsets.
- Also, the number of repetitions is usually larger than in k -fold cross-validation

REPEATED TRAINING/TEST SPLITS

- The number of repetitions is important.
- Increasing the number of subsets has the effect of decreasing the uncertainty of the performance estimates.
- To get stable estimates of performance, it is suggested to choose a larger number of repetitions (say 50–200).
- This is also a function of the proportion of samples being randomly allocated to the prediction set.
- The larger the percentage, the more repetitions are needed to reduce the uncertainty in the performance estimates.

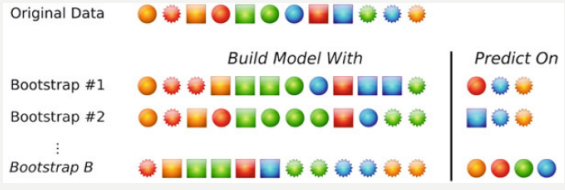
THE BOOTSTRAP

- A bootstrap sample is a random sample of the data taken with replacement (Efron and Tibshirani 1986).
- This means that, after a data point is selected for the subset, it is still available for further selection.
- For a given iteration of bootstrap resampling, a model is built on the selected samples and is used to predict the out-of-bag samples that we shows in next slide's figure.

THE BOOTSTRAP

- In general, bootstrap error rates tend to have less uncertainty than k –fold cross-validation (Efron 1983).
- However, on average, 63.2% of the data points the bootstrap sample are represented at least once, so this technique has bias similar to k -fold cross-validation when k ≈ 2.
- If the training set size is small, this bias may be problematic, but will decrease as the training set sample size becomes larger.

THE BOOTSTRAP



- Each subset is the same size as the original and can contain multiple instances of the same data point.
- Samples not selected by the bootstrap are predicted and used to estimate model performance.

THE BOOTSTRAP

- The "632 method" (Efron 1983) addresses this issue by creating a performance estimate that is a combination of the simple bootstrap estimate and the estimate from re-predicting the training set.
- For example, if a classification model was characterized by its error rate, the 632 method would use

  (0. 632 X simple bootstrap estimate) + (0. 368 X apparent error rate)

THE BOOTSTRAP

- The modified bootstrap estimate reduces the bias, but can be unstable with small samples sizes.
- This estimate can also result in unduly optimistic results when the model severely over-fits the data, since the apparent error rate will be close to zero.
- Efron and Tibshirani (1997) discuss another technique, called the "632+ method," for adjusting the bootstrap estimates.

CASE STUDY: CREDIT SCORING

- The German credit data set is a popular tool for benchmarking machine learning algorithms.
- It contains 1,000 samples that have been given labels of good and bad credit. In the data set, 70% were rated as having good credit.
- Along with these outcomes, data were collected related to credit history, employment, account status, and so on.
- Some predictors are numeric, such as the loan amount.
- However, most of the predictors are categorical in nature, such as the purpose of the loan, gender, or marital status.
- The categorical predictors were converted to "dummy variables" that related to a single category.

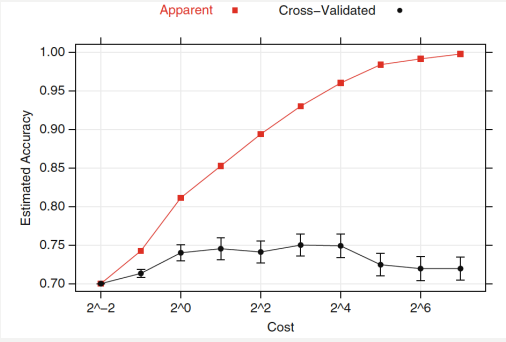
CASE STUDY: CREDIT SCORING

- For example, the applicant's residence information was categorized as either "rent," "own," or "free housing."
- This predictor would be converted to three yes/no bits of information for each category.
- For example, one predictor would have a value of one if the applicant rented and is zero otherwise.
- In all, there were 41 predictors used to model the credit status of an individual.

CHOOSING FINAL TUNING PARAMETERS

- Once model performance has been quantified across sets of tuning parameters, there are several philosophies on how to choose the final settings.
- The simplest approach is to pick the settings associated with the numerically best performance estimates.
- For the credit scoring example, a nonlinear support vector machine model was evaluated over cost values ranging from 2^{-2} to 2^{7}.
- Each model was evaluated using five repeats of 10-fold cross-validation.

CHOOSING FINAL TUNING PARAMETERS



- For each model, cross-validation generated 50 different estimates of the accuracy; the solid points in this figure are the average of these estimates.
- The bars reflect the average plus/minus two-standard errors of the mean.
- The profile shows an increase in accuracy until the cost value is one.

CHOOSING FINAL TUNING PARAMETERS

Resampled accuracy (%)			
Cost	Mean	Std. error	% Tolerance
0.25	70.0	0.0	−6.67
0.50	71.3	0.2	−4.90
1.00	74.0	0.5	−1.33
2.00	74.5	0.7	−0.63
4.00	74.1	0.7	−1.20
8.00	75.0	0.7	0.00
16.00	74.9	0.8	−0.13
32.00	72.5	0.7	−3.40
64.00	72.0	0.8	−4.07
128.00	72.0	0.8	−4.07

- The one-standard error rule would select the simplest model with accuracy no less than 74.3% (75%–0.7%).
- This corresponds to a cost value of 2.
- The "pick-the-best" solution is shown in bold

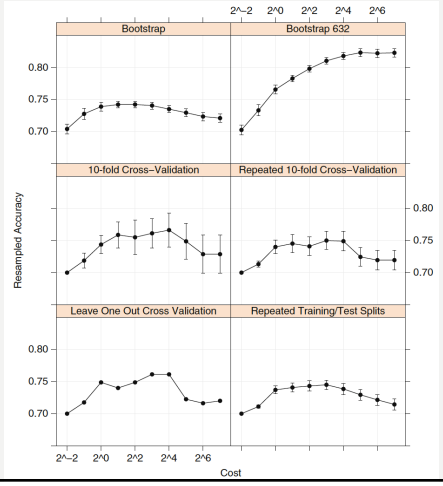
CHOOSING FINAL TUNING PARAMETERS

- In general, it may be a good idea to favor simpler models over more complex ones and choosing the tuning parameters based on the numerically optimal value may lead to models that are overly complicated.
- Other schemes for choosing less complex models should be investigated as they might lead to simpler models that provide acceptable performance (relative to the numerically optimal settings).

CHOOSING FINAL TUNING PARAMETERS

- The "one-standard error" method for choosing simpler models finds the numerically optimal value and its corresponding standard error and then seeks the simplest model whose performance is within a single standard error of the numerically best value.
- This procedure originated with classification and regression trees.
- This technique would find the simplest tuning parameter settings associated with accuracy no less than 74.3% (75%–0.7%).
- This procedure would choose a value of 2 for the cost parameter.

CHOOSING FINAL TUNING PARAMETERS



- A common pattern within the cross-validation methods is seen where accuracy peaks at cost values between 4 and 16 and stays roughly constant within this window.
- The cross-validation techniques estimate the accuracy to be between 74.5% and 76.6%.
- Compared to the other methods, the simple bootstrap is slightly pessimistic.
- Estimating the accuracy to be 74.2% while the 632 rule appears to overcompensate for the bias and estimates the accuracy to be 82.3%.

CHOOSING FINAL TUNING PARAMETERS

- The standard error bands of the simple 10-fold cross-validation technique are larger than the other methods, mostly because the standard error is a function of the number of resamples used (10 versus the 50 used by the bootstrap or repeated splitting).
- The fastest was 10-fold cross-validation, which clocked in at 0.82 min.
- Repeated cross-validation, the bootstrap, and repeated training-test splits fit the same number of models and, on average, took about 5-fold more time to finish.
- LOOCV, which fits as many models as there are samples in the training set, took 86-fold longer and should only be considered when the number of samples is very small.

DATA SPLITTING RECOMMENDATIONS

- There is a strong technical case to be made against a single, independent test set:
  - A test set is a single evaluation of the model and has limited ability to characterize the uncertainty in the results.
  - Proportionally large test sets divide the data in a way that increases bias in the performance estimates.
  - With small sample sizes:
    - The model may need every possible data point to adequately determine model values.
    - The uncertainty of the test set can be considerably large to the point where different test sets may produce very different results.
  - Resampling methods can produce reasonable predictions of how well the model will perform on future samples.

DATA SPLITTING RECOMMENDATIONS

- Here, simple 10-fold cross-validation should provide acceptable variance, low bias, and is relatively quick to compute.
  - If the samples size is small, we recommend repeated 10-fold cross-validation for several reasons: the bias and variance properties are good and, given the sample size, the computational costs are not large.
  - If the goal is to choose between models, as opposed to getting the best indicator of performance, a strong case can be made for using one of the bootstrap procedures since these have very low variance.
  - For large sample sizes, the differences between resampling methods become less pronounced, and computational efficiency increases in importance.

DATA SPLITTING RECOMMENDATIONS

- Varma and Simon (2006) and Boulesteix and Strobl (2009) note that there is a potential bias that can occur when estimating model performance during parameter tuning.
  - The final model is chosen to correspond to the tuning parameter value associated with the smallest error rate.
  - This error rate has the potential to be optimistic since it is a random quantity that is chosen from a potentially large set of tuning parameters.
  - Their research is focused on scenarios with a small number of samples and a large number of predictors, which exacerbates the problem.
- However, for moderately large training sets, our experience is that this bias is small.

CHOOSING BETWEEN MODELS

How do we choose between multiple models?

- This largely depends on the characteristics of the data and the type of questions being answered.
- Predicting which model is most fit for purpose can be difficult.
- Given this, we suggest the following scheme for finalizing the type of model:
  1. Start with several models that are the least interpretable and most flexible, such as boosted trees or support vector machines. Across many problem domains, these models have a high likelihood of producing the empirically optimum results (i.e., most accurate).
  2. Investigate simpler models that are less opaque (e.g., not complete black boxes), such as multivariate adaptive regression splines (MARS), partial least squares, generalized additive models, or naNve Bayes models.
  3. Consider using the simplest model that reasonably approximates the performance of the more complex methods.

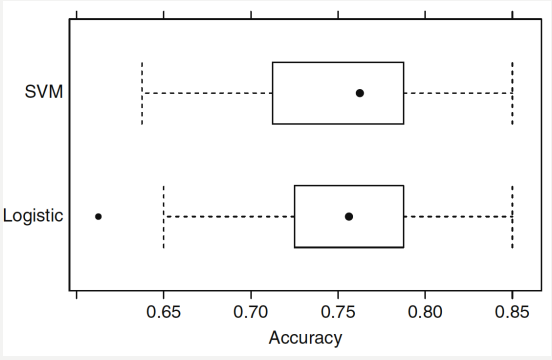
CHOOSING BETWEEN MODELS

- Using this methodology, the modeler can discover the "performance ceiling" for the data set before settling on a model.
- In many cases, a range of models will be equivalent in terms of performance so the practitioner can weight the benefits of different.
- If a more interpretable model, such as a MARS model, yielded similar accuracy, the implementation of the prediction equation would be trivial and would also have superior execution time.

CHOOSING BETWEEN MODELS

- Hothorn et al. (2005) and Eugster et al. (2008) describe statistical methods for comparing methodologies based on resampling results.
- Since the accuracies were measured using identically resampled data sets, statistical methods for *paired comparisons* can be used to determine if the differences between models are statistically significant.
- A paired t -test can be used to evaluate the hypothesis that the models have equivalent accuracies (on average) or, analogously, that the mean difference in accuracy for the resampled data sets is zero.

CHOOSING BETWEEN MODELS



- The 95% confidence interval for this difference was (− 1. 2%, 1 %), indicating that there is no evidence to support the idea that the accuracy for either model is significantly better.
- This makes intuitive sense.
- The resampled accuracies in this figure range from 61.3% to 85%; given this amount of variation in the results, a 0.1% improvement of accuracy is not meaningful.

CHOOSING BETWEEN MODELS

- When a model is characterized in multiple ways, there is a possibility that comparisons between models can lead to different conclusions.
- If the data set includes more events than nonevents, the sensitivity can be estimated with greater precision than the specificity.
- With increased precision, there is a higher likelihood that models can be differentiated in terms of sensitivity than for specificity.

THANK YOU