M5MS03 - Applied Statistics; Coursework 5 - Autumn 2018

Q1. Summarise the key points for Chapter 17 of 'An Introduction to the Bootstrap'. *Pay particular attention to:*

- *The challenges of estimating prediction error.*
- *The specific procedures for bootstrap estimation, noting similarities and differences.*

The chapter discusses various methods for estimating the prediction error of a statistical model [2]. We will summarise its key points in the context of the problem to which it is addressed, which is as follows: given a joint distribution over predictor-response pairs (x, y), a sampled data set $\mathcal{D} = (x_{\mathcal{D}}, y_{\mathcal{D}})$, and a loss function $L(y, \hat{f}(x))$, what is the expected loss of a fitted predictive model $\hat{f} : \mathcal{X} \to \mathcal{Y}$?

Before discussing the content of the chapter in detail, an important point must be mentioned. In practice, we would ideally like to know the expected loss of a model given the training data used to fit it. This quantity is known as the generalization error,

$$\mathrm{GE} = \mathrm{E}[L(Y, \hat{f}(X))|\mathcal{D}] \tag{1}$$

where X and Y are random variables for the feature vector and response respectively. Generalization error is difficult to estimate [3]. Instead, we take its expectation with respect to $\mathcal{D}$ to form the prediction error of the model,

$$\mathrm{PE} = \mathrm{E}[L(Y, \hat{f}(X))] \tag{2}$$

A familiar but inadequate estimator for this quantity is the training error, also known as apparent error, which is defined as

$$\overline{\mathrm{err}} = \frac{1}{n} \sum_{i=1}^{n} L(y_i, \hat{f}(x_i)) \tag{3}$$

where n is the size of the traning data set. An example of a training error estimator is the training residual sum-of-squares (RSS), for which the loss function is squared deviation, $L(y, \hat{y}) = (y - \hat{y})^2$, and the estimate takes the form of a sample mean over the training data set. As Efron says, training error tends to underestimate prediction error because a model is being evaluated on the same dataset that was used to fit it. Additionally, models containing more parameters will tend to have a lower apparent error, but this may simply be a consequence of overfit. Both of these problems are well-illustrated in the context of model selection for linear regression, where the larger of a pair of nested models can be shown to have a training RSS that is less than or equal to that of its smaller counterpart, but may in fact have a higher prediction error.

Since training error is an excessively optimistic estimator for prediction error, alternative estimators have been devised. Some of these attempt to adjust for the dimensionality of the model, while others evaluate the loss function over data sets that are either disjoint or derived from the training data set. Efron mentions that the Akaike Information Criterion accounts for both model dimensionality and the optimism of the training error to yield an approximately unbiased estimator of prediction error:

$$\mathrm{E\,AIC} = \mathrm{E}[2p - 2\log \hat{L}] \approx \mathrm{PE} \tag{4}$$

where $\hat{L}$ denotes the maximum value of the likelihood for a model containing p estimated parameters. The loss function in the prediction error above is the deviance [1]. Caveats of AIC and adjusted residual-squared-error, another form of dimensionality-adjusted error estimator, are that the number of parameters needs to be known, the model must be approximately correct, and the data set must be reasonably large[1].

An alternative way to circumvent issues related to overfit is to evaluate the model's loss on a data set that is separate from the training data set. One of the main topics of Efron's discussion, the cross-validation estimator, does precisely this. Cross-validation involves splitting the available data

[1] The AIC is asymptotically equal to prediction error.

into k tranches ('folds'), evaluating the in-fold loss of a model fit on out-of-fold data, then averaging over the set of k out-of-fold losses. We can express this estimation procedure as

$$\text{CV} = \frac{1}{n}\sum_{i=1}^{n} L(y_i, \hat{f}^{-k(i)}(x_i)) \tag{5}$$

where $\hat{f}^{-k(i)}$ denotes a model fit to a data set that did not contain observation i (along with other observations in the fold k that contains observations i). CV is somewhat similar to test error in the sense that it consists of out-of-sample losses.

Unlike the adjustment-type estimators discussed earlier, cross-validation does not require that a model is roughly correct, nor that is has a known number of parameters. This being said, the model must be fit across a subset of all available data, which may reduce the quality of fit. Equivalently, we could say that an estimate of PE based on this subset-fit model is biased for the PE of the model fit to the full data set. CV may therefore be higher than the actual generalisation error of the model fit to the full data set.

A further complication of cross-validation is that the hyperparameter $k \in \{1, 2, ..., n-1\}$ must be tuned. Depending on the size n of the data set being split, k will affect the bias and variance of the cross-validation estimate, along with the computational expense of evaluating CV [3]. More folds mean that less data is withheld in each fit, implying that the training data sets are all quite similar to one another under each fold[2]. A large-k CV estimate, on the other hand, is sensitive to sampling variability in the full data set. Since the variance of the empirical distribution[3] of the full data set decreases with that set's size, we would expect the a large-k CV estimate on a big dataset to have a smaller variance than one on a small dataset. At the other end of the spectrum, a small-k CV estimate on a small dataset may be biased. This is because a model fit during cross-validation uses only a fraction of the full data set: for small data sets, this makes it substantially worse than a model fit to the full data set (i.e. the model that PE refers to).

In spite of the complications associated with it, cross-validation will give a more realistic estimate of prediction error than training error will. The technique is particularly useful in the context of data sets for which p approaches n ('small' data sets), since it is in these circumstances that overfit is likely to occur.

After discussing cross-validation, Efron moves on to consider various bootstrap methods for estimating prediction error. He covers three methods which we will refer to as the simple, bias-corrected, and .632 bootstrap estimates of prediction error. These methods can be summarized as follows:

- Simple - fit a model to each of B bootstrap samples from the full data set, then evaluate the loss of these models on the original data set. Average over these $n \cdot B$ loss values to obtain a single estimate of the prediction error. This is an optimistic estimate of prediction error since the bootstrap models are evaluated on a data set that has 63%[4] of it data points in common with the data set they were fit with.
- Bias-corrected - evaluate the difference between the bootstrap models' average loss on the original dataset and their average loss on their bootstrap data sets. This is an estimate of the expected difference between the model's prediction error and apparent error, known as its 'optimism'. Add the bootstrap estimate of the optimism to the apparent error of the model fit on the full data set to form a bias-corrected bootstrap estimate of prediction error.
- .632 - fit a model to each of B bootstrap samples from the full data set. Evaluate the loss of these models on the original data set[5], but only for data points that were outside the bootstrap

[2]Consider the case where $k = n$, the data set size. Then a fold consists of 1 data point while the training data set contains the other $n - 1$ datapoints.

[3](as an estimator for the joint distribution of the data)

[4]on average.

[5]Later on this loss a misclassification indicator and the pessimistic error estimate is referred to as a 'leave-one-out' bootstrap estimate of prediction error.

sample used to fit a given model. Average over these losses to form a pessimistic error estimate $\hat{\epsilon}_0$, then take the weighted sum of this and the apparent error for the model fit to the full data set:

$$.362 \cdot \overline{\text{err}} + .632 \cdot \hat{\epsilon}_0 \tag{6}$$

Efron mentions that asymptotically, all of the methods produce the same estimates. Key points he mentions are that:

- Cross-validation is unbiased but can show large variability.
- The simple bootstrap has lower variability but can be downward-biased.
- The 'bias-corrected' bootstrap reduces this downward bias but is still downward-biased.
- The .632 performed best in a few studies.

He concludes the chapter with the following comment:

> *All of the estimates described in this chapter are significant improvements over the apparent error rate. Which is best among these competing methods is not clear.*

As we will see, the experimental results in this report seem to support this conclusion.

Q2. Using the whole of the **`pima`** *dataset and the full regression model, evaluate a 95% bootstrap confidence interval for the coefficient associated with the BMI variable. Contrast this with the normal distribution approximation.*

This question involves logistic regression of the features of `pima` onto the binary response variable `test`. We fit via maximum likelihood a model of the form:

$$\pi_i = g^{-1}(\beta^T x_i) \tag{7}$$

where g is the logistic link function, $g(\pi) = \log \frac{\pi}{1-\pi}$, and β is a vector of real-valued coefficients. x_i is the covariate vector of case i. The fitted value $\hat{\pi}_i$ represents the predicted probability that `test=1`.

We would like to construct a 95% confidence interval for the coefficient β_{bmi}. We will use the paired bootstrap to do this: construct a bootstrap sample by resampling cases (x_i, y_i), then fit the specfied GLM to the bootstrap sample to create a bootstrap replicate $\hat{\beta}^*_{bmi}$ of the `bmi` MLE coefficient. The quantiles of the empricical distribution of $\hat{\beta}_{bmi} - \hat{\beta}^*_{bmi}$ can then be used to create a confidence interval for β_{bmi}.

Before discussing the details of the bootstrap confidence intervals, we explain why resampling cases is better-motivated than resampling residuals, since the residual bootstrap is a popular alternative for regression problems. The validity of the residual bootstrap is contingent on the resampled residuals having constant variance with respect to the covariates. A priori, there is no reason to believe that this will be the case for the chosen model and `pima` data set. The paired bootstrap is therefore a more reasonable choice. In a similar vein, there are many variants of bootstrap confidence intervals. We will be using the basic bootstrap interval because it is straightforward to implement and relies on fewer assumptions than Efron's percentile method[6].

We derive the basic bootstrap confidence interval. Briefly let β denote a scalar parameter and $\hat{\beta}$ an estimator for it. Given level α we seek constants c_1 and c_2 such that

$$P(c_1 \leqslant \hat{\beta} - \beta \leqslant c_2) = 1 - \alpha \tag{8}$$

The bootstrap plug-in principle can be applied to obtain:

$$P(c_1^* \leqslant \hat{\beta}^* - \hat{\beta} \leqslant c_2^*) = 1 - \alpha \tag{9}$$

[6]Specifically, it does not assume that the bootstrap estimates have a symmetric distribution.

This equation is satisfied by a set of (c_1^*, c_2^*) values. Since we want just one confidence interval, we introduce the requirements

$$P(c_1^* \leqslant \hat{\beta}^* - \hat{\beta}) = P(\hat{\beta}^* - \hat{\beta} \leqslant c_2^*) = 1 - \frac{\alpha}{2} \tag{10}$$

These conditions imply that

$$P(\hat{\beta}^* - c_2^* \leqslant \hat{\beta} \leqslant \hat{\beta}^* - c_1^*) = 1 - \alpha \tag{11}$$

i.e. our bootstrap estimate for the confidence interval should be $(\hat{\beta} - c_2^*, \hat{\beta} - c_1^*)$. Percentile $\frac{\alpha}{2}$ of the distribution of $\hat{\beta}^* - \hat{\beta}$ corresponds to c_1^*, whereas percentile $1 - \frac{\alpha}{2}$ is c_2^*.

Let β once again refer to the coefficient vector in the GLM. Asymptotically and under the assumption that the data were generated according to a model parameterized by β, the maximum-likelihood estimator has a multivariate normal distribution with covariance matrix $\mathcal{J}^{-1}$, where $\mathcal{J}$ denotes the Fisher information matrix. It follows that the asymptotic standard error of $\hat{\beta}_i$ is $\sqrt{\mathcal{J}_{ii}^{-1}}$. This is the quantity returned by R's `summary` function as `Std. Error`[7].

The code below implements the procedures described above to obtain bootstrap and normal-theory 95% confidence intervals for β_{bmi}. Note the confidence intervals printed on the next page.

```
set.seed(3298760)
library(faraway)
data(pima)
Pima    <- pima

# Allocate NA values
zero_names <- c('glucose', 'diastolic', 'triceps', 'insulin', 'bmi')
Pima[zero_names][Pima[zero_names] == 0] <- NA
Pima$test <- as.factor(Pima$test)

# Exclude incomplete cases
Pima <- Pima[complete.cases(Pima), ]

# GLM formula
f <- formula(test ~ 1 + insulin + bmi + diabetes + glucose +
                    age + pregnant + diastolic + triceps)

# Bootstrap machinery
getBootstrapSample <- function(Z, N=nrow(Z)) Z[sample(1:N, size=N, replace=T), ]
getBootstrapRep <- function(Z){
                     Z_star  <- getBootstrapSample(Z)
                     lg_star <- glm(f, data=Z_star, family=binomial(link='logit'))
                     b_star  <- as.numeric(lg_star$coef['bmi'])
                     return(b_star)
                   }

getBootstrapCI <- function(Z, B=9999, alpha=0.05){
                    lg_hat  <- glm(f, data=Z, family=binomial(link='logit'))
                    b_hat   <- as.numeric(lg_hat$coef['bmi'])
                    b_star  <- sapply(1:B, function(t) getBootstrapRep(Z))
                    c1_star <- quantile(b_star - b_hat, alpha/2, names=F)
                    c2_star <- quantile(b_star - b_hat, 1 - alpha/2, names=F)
                    return(c(mle=b_hat, lwr=b_hat - c2_star, upr=b_hat - c1_star))
```

[7] Strictly speaking, R returns the square-root of the diagonal element of the inverse of the observed information matrix. This matrix is an estimator that converges in probability (with n) to the Fisher information.

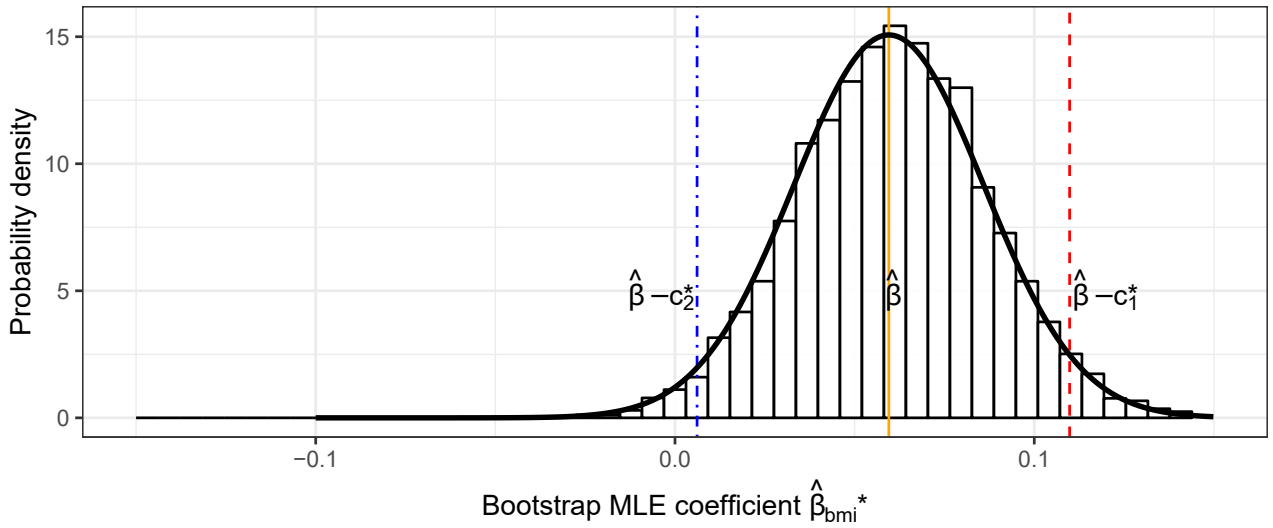

Figure 1: Q2. Histogram of $N = 9999$ bootstrap maximum-likelihood estimates $\hat{\beta}^*_{bmi}$, with a normal distribution with standard deviation equal to their standard error, $\sigma = 0.026$, superimposed. The asymptotic standard error for $\hat{\beta}$ is 0.025. We can see that the asymptotic and bootstrap confidence intervals agree remarkably closely.

```
                  }

# Asymptotic MLE confidence interval
getNormalCI <- function(Z, alpha=0.05){
                  lg_hat <- glm(f, data=Z, family=binomial(link='logit'))
                  se     <- summary(lg_hat)$coef[3, 2]
                  b_hat  <- lg_hat$coef[3]
                  return(b_hat + qnorm(c(alpha/2, 1-alpha/2))*se)
                }



round(getBootstrapCI(Pima), 2) # Basic bootstrap 95% confidence interval

##  mle  lwr  upr
## 0.07 0.01 0.13

round(getNormalCI(Pima), 2) # Normal theory 95% confidence interval

## [1] 0.02 0.12
```

We can see that the asymptotic confidence interval is in close agreement with the basic bootstrap interval. The distribution of the bootstrap replicates $\hat{\beta}^*_{bmi}$, shown in Figure 1, provides further evidence that the asymptotic-theory result is applicable to this problem. Assume that the empirical distribution function $\hat{F}$ for the complete-case `pima` data set is a good approximation of the actual distribution function F. Under this assumption, the distribution of bootstrap MLEs $\hat{\beta}^*(\hat{F})$ accurately approximates the distribution of the real-world MLEs $\hat{\beta}(F)$. The distribution of bootstrap MLEs shown in Figure 1 is approximately normal with variance $\approx \sqrt{\mathcal{J}_{ii}}$, therefore provided $\hat{F}$ is a good estimate of F, the normal theory confidence intervals are appropriate. Given the size of the data set, the assumption $\hat{F} \approx F$ seems reasonable.

Q3. Select one of the bootstrap procedures from Q1 for estimating the error rate of the

diagnostic test. Contrast this estimate with the result from Coursework 4. Discuss the pros and cons of the different approaches.

The 'diagnostic test' is our GLM, which we can use to predict `test`. In Coursework 4, we estimated the misclassification rate of this model by splitting the data into a training set and a testing set, fitting the model to the training set, and evaluating the fit model against the testing set. We made predictions by setting `test=1` when the probability predicted by the model was greater than 0.5. The code below recomputes the misclassification rate on the testing set in the same way as done for that coursework.

```
  set.seed(23470)
  shuffled_ix <- sample(1:nrow(Pima), size=nrow(Pima), replace=F)
  training_ix <- shuffled_ix[1:292]
  test_ix     <- shuffled_ix[293:392]

  Pima.full   <- glm(f, data=Pima[training_ix, ],
                     family=binomial(link='logit'))

  pred_p       <- predict(Pima.full,  type='response')
  ground_truth <- as.logical(as.numeric(Pima$test[training_ix])-1)
  round(mean((pred_p>0.5)!=ground_truth), 3) # Apparent error
```

```
## [1] 0.178
```

```
  pred_p       <- predict(Pima.full, newdata=Pima[test_ix, ], type='response')
  ground_truth <- as.logical(as.numeric(Pima$test[test_ix])-1)
  round(mean((pred_p>0.5)!=ground_truth), 3) # Test-set misclass. rate
```

```
## [1] 0.29
```

To evaluate these vales as estimates of the prediction error, we consider three further methods for estimating the misclassification rate: cross-validation, the simple bootstrap error, and the .632 bootstrap method. Start with cross-validation. As has been discussed, the optimal number of folds k for a cross-validation estimate of prediction error depends on the size of the data set under study[8]. In this case, the data set contains 392 observations. If we set $k = 10$, then at each step of the cross-validation algorithm the logistic regression model would be fit using 353 data points and evaluated on 39 data points. The performance of a model fit using 353 data points is likely to be an unbiased estimate for the performance of a model fit using 392 data points. Furthermore, this size still permits non-negligible sampling variability within the CV training data sets. $k = 10$ is also recommended in the literature as a suitable bias-variance compromise for the cross-validation estimator [3]. The code below implements a cross-validation estimate.

```
  set.seed(78934)

  # 10-fold cross-validation
  k    <- 10
  N    <- nrow(Pima)
  Pima <- Pima[sample(1:N, size=N, replace=F), ] # Shuffle rows
  n    <- floor(N/k)       # No. elements per fold
  r    <- nrow(Pima) %% n # Surplus elements in final fold
  fold_ix   <- lapply(0:(k-1), function(j) return((j*n + 1):((j+1)*n)))
  fold_ix[k][[1]] <- c(fold_ix[k][[1]], (N-r+1):N)
  cv_errors <- 0

  for(j in 1:k){ # Iterate over collection of index sets
```

[8] It also depends on the dimensionality of the feature space, but a discussion of this would go beyond this report's scope.

```
    test_ix <- fold_ix[j][[1]]
    # Fit model
    Pima.full   <- glm(f, data=Pima[-test_ix, ],
                       family=binomial(link='logit'))
    # Make out-of-bag predictions
    pred_p       <- predict(Pima.full, newdata=Pima[test_ix, ], type='response')

    # Accumulate missclass. on out-of-bag observations
    ground_truth <- as.logical(as.numeric(Pima$test[test_ix])-1)
    cv_errors    <- cv_errors + sum((pred_p>0.5)!=ground_truth)
  }

  # Divide by the size of the data set
  cv_error <- cv_errors/N
  round(cv_error, 3) # CV misclass. rate
```

```
## [1] 0.222
```

We can see that the cross-validation misclassification rate (0.22) is in-between the apparent error (0.18) and the test error (0.29). This may suggest that the test error was unduly pessimistic as a result of test-set sampling variability.

We now consider the bootstrap estimates of prediction error. Start with the simple bootstrap: resample the data set, fit the model to the resampled data set, evaluated the model's misclassification rate on the original data set, then average these rates across the bootstrap replicates.

```
  getBootMR <- function(Z){
                bootZ    <- getBootstrapSample(Z=Z)
                lg_star  <- glm(f, data=bootZ,
                                family=binomial(link='logit'))
                pred_p   <- predict(lg_star, newdata=Z, type='response')
                ground_truth  <- as.logical(as.numeric(Z$test)-1)
                misclass_rate <- sum((pred_p>0.5)!=ground_truth)/nrow(Z)
                return(misclass_rate)
              }
  B      <- 9999
  bootMR <- sapply(1:B, function(t) getBootMR(Pima))
  round(mean(bootMR), 3) # Simple bootstrap estimate of prediction error
```

```
## [1] 0.211
```

The simple bootstrap error is slightly below the cross-validation error (0.22) and is above the apparent error (0.18). These differences are surprising because the bootstrap data sets used to fit the model have a large number of data points in common with the original data set. We might have expected, therefore, that the simple bootstrap estimate would be closer to to the apparent error and be slightly too optimistic.

To compute the .632 bootstrap error, were must first estimate the leave-one-out bootstrap estimate of prediction error. This is similar to the simple bootstrap error but the replicate misclassification rate is evaluated on out-of-sample data points only.

```
  getBootLOOMR <- function(Z, N=nrow(Z)){ # leave-one out
                    boot_ix  <- sample(1:N, size=N, replace=T)
                    bootZ    <- Z[boot_ix, ]
                    lg_star  <- glm(f, data=bootZ,
                                    family=binomial(link='logit'))
                    pred_p   <- predict(lg_star, newdata=Z[-boot_ix, ],
```

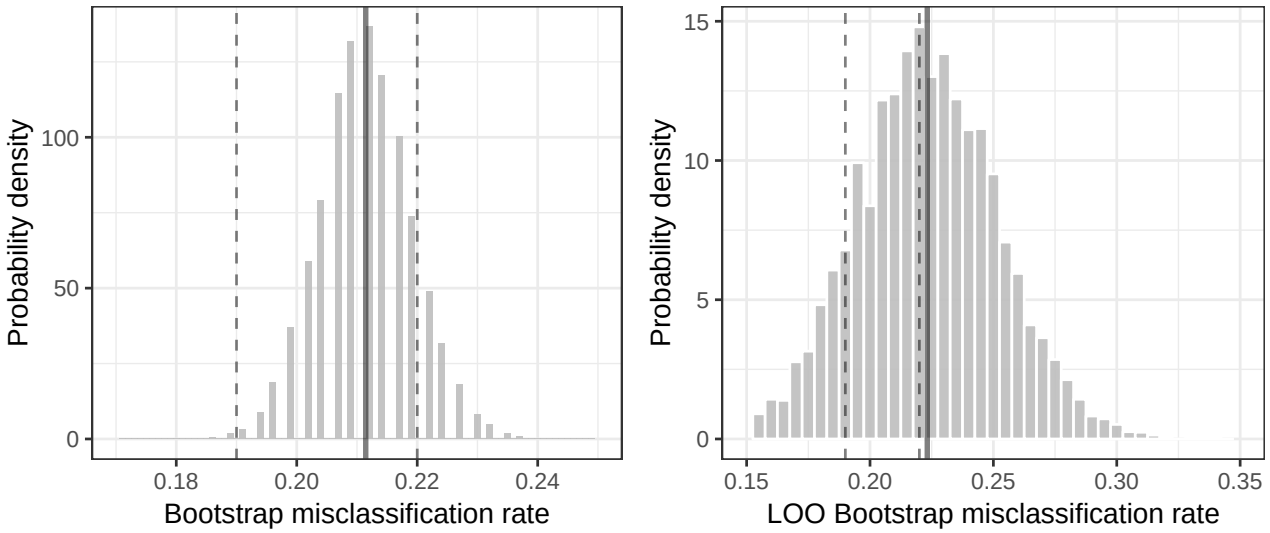

Figure 2: Histograms of bootstrap misclassification rates across $B = 9999$ bootstrap samples for simple (left) and leave-one-out (right) methods. The left dashed line is the apparent error of the logistic regression model. The right dashed line is the cross-validation estimate of prediction error. The solid line is the sample mean of the bootstrap misclassification rates.

```
                                    type='response')
    ground_truth  <- as.logical(as.numeric(Z$test[-boot_ix])-1)
    misclass_rate <- sum((pred_p>0.5)!=ground_truth)/length(pred_p)
    return(misclass_rate)
  }

B <- 9999
bootLOOMR <- sapply(1:B, function(t) getBootLOOMR(Pima))
round(mean(bootLOOMR), 3)
```

```
## [1] 0.223
```

The leave-one-out (LOO) bootstrap misclassification rate (0.22) agrees closely with the cross-validation error (0.22). This suggests that the cross-validation estimate may be slightly pessimistic, and that the bootstrap models are of a similar quality to the cross-validation models. Both are evaluated on out-of-sample data points, but they are fit on slightly different data sets. The CV models have access to 352 unique data points, whereas the bootstrap models have access to, on average, $.632 \cdot 392 \approx 248$ unique data points. Reassuringly, the LOO bootstrap error (0.22) is larger than the simple bootstrap error (0.21).

The LOO bootstrap error rate allows us to evaluate the .632 estimate of prediction error: $.368 \cdot .178 + .632 \cdot .223 = 0.206$. Efron cautiously indicates that this is the best of the bootstrap estimators for prediction error. This being said, its value is slightly smaller than that of the simple bootstrap, which is likely to be too optimistic.

The discrepancies between the bootstrap estimates could be the subject of an extensive analysis. Crucially, all estimates are larger than the training-set error and smaller than the test-set error. While the former result could have been anticipated, the latter could not: it is difficult to know the amount of sampling variability that the test set is subject to. The bootstrap and cross-validation estimates provide evidence that the expected misclassification rate of the diagnostic tool is likely to be between 20-22%, as opposed to the 29% or 18% suggested by the test-set and training-set error rates respectively.

A broad conclusion that can be drawn from this report is that making a small amount of effort

to accurately estimate prediction error is categorically worthwhile. This is especially true in the context of model-selection, where prediction error is the key measure of performance, and when a model's predictive performance has dramatic real-world consequences, such as in clinical diagnostics or algorithmic trading.

References

[1] H. Akaike. "A new look at the statistical model identification". In: *IEEE Transactions on Automatic Control* 19.6 (1974), pp. 716–723. ISSN: 0018-9286.

[2] B. Efron and R. J. Tibshirani. *An Introduction to the Bootstrap*. New York: Chapman & Hall, 1993.

[3] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning*. Springer Series in Statistics. New York, NY, USA: Springer New York Inc., 2001.