# Forming Multiple Confidence Intervals in Monte Carlo Data

Norm Matloff

## Table of contents

Author bio

In developing/choosing a model, we may perform cross-validation, with many replicates, in order to compare multiple algorithms, multiple sets of hyperparameters and so on. To make our analysis statistically valid, we can form confidence intervals (CIs).

However, if we form many CIs, say at 95% level each, their overall coverage probability may be much lower than 95%. This is the *multiple inference* (MI) or *simultaneous inference* problem. In this document, we discuss remedies in the model-selection context.

Some analysts in some applications may not consider this to be a "problem." This is a philosophical issue, not pursued here.

The MI problem has been extremely well studied, resulting in myriad methods. Here we employ two of the most well-known methods, the Bonferroni Inequality and Scheffe's Method.

See Jason Hsu, *Multiple Comparisons: Theory and methods.*

Note that our focus is on CIs, not hypothesis tests. We strongly recommend against the latter approach.

## Motivating Example

```
library(qeML)
data(svcensus)
head(svcensus)
```

This is US census data. Let's predict gender.

```
logitAcc <- qeLogit(svcensus,'gender')$testAcc
rfAcc <- qeRFranger(svcensus,'gender')$testAcc
xgbAcc <- qeXGBoost(svcensus,'gender')$testAcc
c(logitAcc,rfAcc,xgbAcc)
```

Several points to note:

- The **qeML** functions automatically do cross-validation, via an argument **holdout**; here we take the default value.
- The functions return S3 objects, one of whose components is prediction accuracy on the test data, **testAcc**, in this case the probability of misclassification..
- Since the random number seed is not reset, each of the three algorithms is using different training sets and different test sets from each other. This makes them statistically independent. The alternative (not necessarily better or worse) would be to insert, say,

  ```
  set.seed(9999)
  ```

  before each of the three calls.
- Since the holdout set is random, we should be performing each of the three calls many times, and compute three averages, say

  ```
  logitAccs <-
     replicate(50,qeLogit(svcensus,'gender')$testAcc)
  rfAccs <-
     replicate(50,qeRFranger(svcensus,'gender')$testAcc)
  xgbAccs <-
     replicate(50,qeXGBoost(svcensus,'gender')$testAcc)
  ```

```
accs <- cbind(logitAccs,rfAccs,xgbAccs)
colMeans(accs)
```

## Review: Confidence Intervals, Standard Errors

To set the stage, let's review the statistical concepts of *confidence interval* and *standard error*. Say we have an estimator $\hat{\theta}$ of some population parameter $\theta$, e.g. $\bar{X}$ for a population mean $\mu$.

Loosely speaking, the term *standard error* of is our estimate of $\sqrt{Var(\hat{\theta})}$. More precisely, suppose that $\hat{\theta}$ is asymptotically normal. The standard error is an estimate of the standard deviation of that normal distribution. For this reason, it is customary to write $AVar(\hat{\theta})$ rather than $Var(\hat{\theta})$.

A, say 95%, confidence interval (CI) for $\mu$ is then

$$\hat{\theta} \pm 1.96 \text{ SE}(\hat{\theta})$$

where we denote the standard error of $\hat{\theta}$ by $\text{SE}(\hat{\theta})$.

The 95% figure means that of all possible samples of the given size from the population, 95% of the resulting confidence intervals will contain $\theta$. In many cases, the 95% figure is only approximate, stemming from a derivation that uses the Central Limit Theorem.

In general, for confidence level $1 - \alpha$, replace 1.96 by $z_\alpha$, the $1 - \alpha/2$ quantile of the N(0,1) distribution, Then our CI is

$$\hat{\theta} \pm z_\alpha \text{SE}(\hat{\theta}) \tag{1}$$

Examples of finding $z_\alpha$:

```
> qnorm(0.975)
[1] 1.959964  # for 95% CI
> qnorm(0.995)  # for 99% CI
[1] 2.575829
```

! Note Regarding Sensitivity of Phrasing

There is a bit of drama in this word *contain* in the phrase "will contain $\theta$." Instead of saying the intervals *contain* $\theta$, why not simply say $\theta$ is *in* the intervals? Aren't these two descriptions equivalent in terms of English? Of course they are.

But many instructors of statistics classes worry that students will take the description based on "in" to mean that $\theta$ is the random quantity, when in fact the CI is random (random center, random radius) and $\theta$ is fixed (though unknown). The instructors thus insist on the more awkward phrasing "contain," so as to avoid students misunderstanding. Indeed some instructors would contend that use of the word *in* is itself just plain incorrect.

My own view is that in some cases the word *in* is clearer (and certainly correct in any case), and that it is better to add a warning about what is random/nonrandom than engage in awkward phrasing.

### Example: Logistic Regression Coefficients

```r
suppressPackageStartupMessages(library(qeML))
data(svcensus)
logitOut <- qeLogit(svcensus,'gender',yesYVal='female')
summary(logitOut$glmOuts[[1]])
```

```

Call:
glm(formula = yDumm ~ ., family = binomial, data = tmpDF)

Coefficients:
               Estimate Std. Error z value Pr(>|z|)
(Intercept)  -5.743e-01  9.454e-02  -6.075 1.24e-09 ***
age           5.823e-03  1.544e-03   3.771 0.000162 ***
educ16       -5.420e-01  1.174e-01  -4.617 3.89e-06 ***
educzzzOther -9.860e-02  4.358e-02  -2.262 0.023672 *
occ101       -3.517e-01  4.784e-02  -7.351 1.96e-13 ***
occ102       -3.714e-01  4.497e-02  -8.257  < 2e-16 ***
```

```
occ106        3.699e-01  9.916e-02   3.730 0.000192 ***
occ140       -8.790e-01  1.047e-01  -8.395  < 2e-16 ***
occ141       -1.463e+00  7.325e-02 -19.974  < 2e-16 ***
wageinc      -6.566e-06  5.343e-07 -12.289  < 2e-16 ***
wkswrkd       1.330e-03  1.289e-03   1.031 0.302314
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 21251  on 19089  degrees of freedom
Residual deviance: 20331  on 19079  degrees of freedom
AIC: 20353

Number of Fisher Scoring iterations: 4
```

So a 95% CI for the coefficient for occupation 141 is

$$-1.46 \pm 1.96 \times 0.07$$

## The Bonferroni Inequality

This one is the simplest and most convenient.