Stat 5421 Lecture Notes: Simple Chi-Square Tests for Contingency Tables

Charles J. Geyer

August 24, 2022

Contents

1 License

2 One-Way Contingency Table

The data set read in by the R function `read.table` below simulates 6000 rolls of a fair die (singular of dice).

```
foo <- read.table(
    url("http://www.stat.umn.edu/geyer/5421/mydata/multi-simple-1.txt"),
    header = TRUE)
print(foo)
```

```
##      y num
## 1 1038   1
## 2  964   2
## 3  975   3
## 4  983   4
## 5 1035   5
## 6 1005   6
```

2.1 Goodness of Fit Test, Completely Specified Null Hypothesis

We test the hypothesis that all six cells of the contingency table have the same probability (null hypothesis) versus that they are different (alternative hypothesis).

2.1.1 Pearson Chi-Squared Test Statistic

```
out <- chisq.test(foo$y)
print(out)
```

```
##
##  Chi-squared test for given probabilities
##
## data:  foo$y
## X-squared = 4.904, df = 5, p-value = 0.4277
```

The default test statistic for the R function `chisq.test` is the Pearson chi-squared test statistic

$$X^2 = \sum_{\text{all cells of table}} \frac{(\text{observed} - \text{expected})^2}{\text{expected}}. \tag{1}$$

We write this in more mathematical notation as follows. Let I denote the index set for the contingency table, which is one-dimensional in this example, but may have any dimension in general, so making our subscript i range over an abstract set rather than a range of numbers means we do not have to have a different formula for every dimension of table. Let x_i denote the observed count for cell i, $\tilde{\pi}_i$ the cell probability for the i-cell under the null hypothesis, and n the sample size (so the random vector having components x_i is multinomial with sample size n and parameter vector having components $\tilde{\pi}_i$). Then the Pearson test statistic is

$$X^2 = \sum_{i \in I} \frac{(x_i - n\tilde{\pi}_i)^2}{n\tilde{\pi}_i}. \tag{2}$$

The Pearson chi-square test statistic is a special case of the Rao test statistic (also called score test and Lagrange multiplier test, see the likelihood handout for its definition).

Consequently, its asymptotic distribution is chi-square with degrees of freedom which is the difference of dimensions of the models being compared. Here the null model is completely specified (no adjustable parameter) so has dimension zero, and the multinomial has $k - 1$ degrees of freedom if the table has k cells because probabilities must sum to one, that is $\sum_{i \in I} \pi_i = 1$ so specifying $k - 1$ parameters determines the last one. Hence the degrees of freedom of the asymptotic chi-square distribution is $k - 1$. Here $k = 6$ so the degrees of freedom is 5, which is what the printout of the `chisq.test` function says.

2.1.2 Likelihood Ratio Test Statistic

We can also do a Wilks test (also called likelihood ratio test).

```
Gsq <- 2 * sum(out$observed * log(out$observed / out$expected))
print(Gsq)
```

```
## [1] 4.894725
```

```
Gsq.pval <- pchisq(Gsq, out$parameter, lower.tail = FALSE)
print(Gsq.pval)
```

```
## [1] 0.4288628
```

The probability mass function (PMF) for the multinomial distribution is

$$f_\pi(\mathbf{x}) = \binom{n}{\mathbf{x}} \prod_{i \in I} \pi_i^{x_i},$$

where the thingy with the round brackets is a multinomial coefficient. Since that does not contain the parameter, we can drop it in the likelihood

$$L(\pi) = \prod_{i \in I} \pi_i^{x_i},$$

so the log likelihood is

$$l(\pi) = \sum_{i \in I} x_i \log(\pi_i)$$

(in this handout we are using the convention that boldface denotes a vector and normal font it components, so π is the vector having components π_i). If $\hat{\pi}$ and $\tilde{\pi}$ are the maximum likelihood estimates (MLE) in the alternative and null hypotheses, respectively, then the Wilks statistic is

$$G^2 = 2\big[l(\hat{\pi}) - l(\tilde{\pi})\big] = 2 \sum_{i \in I} x_i \log\left(\frac{\hat{\pi}_i}{\tilde{\pi}_i}\right) \tag{3}$$

We already know the MLE in the null hypothesis (completely specified, all cell probabilities the same). To determine the MLE in the alternative hypothesis, we need to solve the likelihood equations for the multinomial distribution, and this is a bit tricky because of the degeneracy of the multinomial. We have to write one parameter as a function of the others to get down to $k - 1$ freely adjustable variables.

Fix $i^* \in I$ and eliminate that variable writing $I^* = I \setminus \{i^*\}$ so

$$l(\pi) = \sum_{i \in I^*} x_i \log(\pi_i) + x_{i^*} \log\left(\sum_{i \in I^*} \pi_i\right)$$

so for $i \in I^*$

$$\frac{\partial l(\pi)}{\partial \pi_i} = \frac{x_i}{\pi_i} - \frac{x_{i^*}}{1 - \sum_{i \in I^*} \pi_i} = \frac{x_i}{\pi_i} - \frac{x_{i^*}}{\pi_{i^*}}$$

Setting derivatives to zero and solving for the parameters gives

$$\pi_i = x_i \cdot \frac{\pi_i^*}{x_{i^*}}, \qquad i \in I^*. \tag{4}$$

In order to make progress we have to figure out what π_{i^*} is. The only facts we have at our disposal to do that is that probabilities sum to one and cell counts sum to n. So we first note that (4) trivially holds when $i = i^*$, so summing both sides of (4) over all i gives

$$\sum_{i \in I} \pi_i = \frac{\pi_i^*}{x_{i^*}} \sum_{i \in I} x_i$$

or

$$1 = \frac{\pi_i^*}{x_{i^*}} \cdot n$$

or

$$\pi_i^* = \frac{x_{i^*}}{n}$$

and plugging this into (4) and writing $\hat{\pi}_i$ rather than π_i because our solution is the MLE gives

$$\hat{\pi}_i = \frac{x_i}{n}, \qquad i \in I. \tag{5}$$

So we have found that the unrestricted MLE for the multinomial distribution is just the observed cell counts divided by the sample size. This is, of course, the obvious estimator, but we didn't know it was the MLE until we did the math.

We now also rewrite (3) gratuitously inserting n's in the numerator and denominator so it uses the quantities used in the Pearson statistic

$$\begin{aligned} G^2 &= \sum_{i \in I} x_i \log\left(\frac{n\hat{\pi}_i}{n\tilde{\pi}_i}\right) \\ &= \sum_{\text{all cells of table}} \text{observed} \cdot \log\left(\frac{\text{observed}}{\text{expected}}\right) \end{aligned} \tag{6}$$

And that explains the calculation done in R.

The tests are asymptotically equivalent so it is no surprise that the test statistics are very similar (4.904 and 4.8947), as are the P-values (0.4277 and (0.4289). Since the P-values are not small, the null hypothesis is accepted. The die seems fair.

2.2 Hypothesis Tests with Composite Null Hypotheses

The data set read in by the R function `read.table` below simulates 6000 rolls of an unfair die of the type known as six-ace flats. The one and six faces, which are on opposite sides of the die, are shaved slightly so the other faces have smaller area and smaller probability.

```
foo <- read.table(
    url("http://www.stat.umn.edu/geyer/5421/mydata/multi-simple-2.txt"),
    header = TRUE)
foo
```

```
##      y num
## 1 1047   1
## 2 1017   2
## 3  951   3
## 4 1004   4
## 5  952   5
## 6 1029   6
```

2.2.1 Naive Goodness of Fit Test

(This test does not actually belong in this section. It has a simple null hypothesis.)

We start by testing the hypothesis that all six cells of the contingency table have the same probability (just like in the preceding section).

```
y <- foo$y
### Pearson chi-square test statistic
out <- chisq.test(y)
print(out)
```

```
##
##  Chi-squared test for given probabilities
##
## data:  y
## X-squared = 8.06, df = 5, p-value = 0.153
```

```
### likelihood ratio test statistic
Gsq <- 2 * sum(out$observed * log(out$observed / out$expected))
print(Gsq)
```

```
## [1] 8.094506
```

```
pchisq(Gsq, out$parameter, lower.tail = FALSE)
```

```
## [1] 0.1511036
```

The P-values are lower than before, so perhaps slightly suggestive that the null hypothesis is false, but still much larger than 0.05 (not that your humble author is suggesting that 0.05 is the dividing line between statistical significance and statistical non-significance, but these are not very low P-values). Hence we could again conclude the die seems fair. But now that would be wrong because we haven't even fit the six-ace hypothesis yet!

2.2.2 Goodness of Fit Test of Six-Ace Flats Hypothesis

2.2.2.1 Maximum Likelihood Estimate

A calculation similar to the one in the preceding section (which we will mercifully not do, it will eventually be justified by the observed-equals-expected principle) says that the MLE for the six-ace hypothesis estimates

the cell probabilities for the six and ace cells to be the average of the observed cell frequencies for these two cells

$$\hat{\pi}_1 = \hat{\pi}_6 = \frac{x_1 + x_6}{2n}$$

(because they are assumed to have the same area) and similarly for the other cells

$$\hat{\pi}_2 = \hat{\pi}_3 = \hat{\pi}_4 = \hat{\pi}_5 = \frac{x_2 + x_3 + x_4 + x_5}{4n}$$

(because they are assumed to have the same area).

This is the MLE for the six-ace flats model. The dimension of this model is one because if we know $\hat{\pi}_1$, then we can also figure out all of the other cell probabilities from the constraints that probabilities sum to one and the equality constraints above. Hence this model has dimension one.

```
nrolls <- sum(y)
pi.hat <- rep(NA, 6)
pi.hat[c(1, 6)] <- sum(y[c(1, 6)]) / 2 / nrolls
pi.hat[- c(1, 6)] <- sum(y[- c(1, 6)]) / 4 / nrolls
print(pi.hat)
```

```
## [1] 0.1730 0.1635 0.1635 0.1635 0.1635 0.1730
```

2.2.2.2 Pearson Chi-Squared Test

One might think that we could just use R function `chisq.test` as before.

```
out <- chisq.test(y, p = pi.hat)
print(out)
```

```
##
##  Chi-squared test for given probabilities
##
## data:  y
## X-squared = 3.7911, df = 5, p-value = 0.5799
```

But this is incorrect, because R function `chisq.test` does not know we estimated one parameter in the null hypothesis and this changes the degrees of freedom. And there is no way to tell it about this.

It does correctly calculate the Pearson chi-squared test statistic.

```
tstat <- out$statistic
print(tstat)
```

```
## X-squared
##  3.791136
```

But the degrees of freedom is one less than what it thinks because we estimated one parameter in the null hypothesis. In general, if we estimated k parameters, then the degrees of freedom would be k less.

So the P-value for the Pearson chi-squared test is

```
df <- out$parameter - 1
print(df)
```

```
## df
##  4
```

```
pchisq(tstat, df = df, lower.tail = FALSE)
```

```
## X-squared
## 0.4350099
```

2.2.2.3 Likelihood Ratio Test

We can also do a likelihood ratio test instead.

```
expected <- out$expected
print(expected)
```

```
## [1] 1038  981  981  981  981 1038
```

```
observed <- y

tstat <- 2 * sum(observed * log(observed / expected))
print(tstat)
```

```
## [1] 3.789174
```

```
pchisq(tstat, df = df, lower.tail = FALSE)
```

```
## [1] 0.4352892
```

We used (6) as the formula for the likelihood ratio test statistic.

2.2.3 Tests of Model Comparison

The preceding two sections were somewhat unsatisfactory. They seem to say that both null hypothesis (fair dice and six-ace flats (unfair dice)) fit the data. But they can't both be right. Of course the test statistics seem to say that the six-ace flats model fits even better than the fair dice model. But that is not how hypothesis tests work.

To really get a handle on these data, we need a better hypothesis test.

Goodness of fit tests always have anything at all as the alternative hypothesis.

General tests of model comparison can compare any two models, so long as they are nested (the null hypothesis is a submodel of the alternative hypothesis). Here the fair dice model is the special case of the six-ace flats model when $\alpha = 1/6$. So these models are nested. (For more on the concept of nested models, see the notes on likelihood.)

So now we want to compare these two models. One with MLE cell probabilities `pi.hat` and the other with MLE cell probabilities

```
pi.twiddle <- rep(1 / 6, 6)
```

The difference in number of (freely adjustable) parameters of the two models is one (one parameter in six-ace flats, zero parameters in fair dice). So the test statistics (Wald, Wilks, and Rao) are approximately chi-square on one degree of freedom.

2.2.3.1 Likelihood Ratio Test

The likelihood ratio test is

```
### likelihood ratio test statistic
Gsq <- 2 * sum(y * log(pi.hat / pi.twiddle))
print(Gsq)
```

```
## [1] 4.305331
```

```
Gsq.pval <- pchisq(Gsq, 1, lower.tail = FALSE)
print(Gsq.pval)
```

```
## [1] 0.03799309
```

We used (3) as the formula for the likelihood ratio test statistic.

Now $P = 0.038$ is moderately strong evidence that the null hypothesis (fair die) is false and the alternative (six-ace flat) is true. The moral of the story: you cannot conclude anything about a hypothesis (model) until you have actually fit that hypothesis (model), and, if you want to compare two models, then use a test of model comparison comparing them.

2.2.3.2 Rao Test

The Rao test depends on what we take to be the parameter of the six-ace hypothesis, say the probability α for the six-ace cells. If the cell probabilities are $(\alpha, \beta, \beta, \beta, \beta, \alpha)$, then

$$\beta = \frac{1 - 2\alpha}{4}$$

and

$$l(\alpha) = (x_1 + x_6) \log(\alpha) + (x_2 + x_3 + x_4 + x_5) \log(\beta)$$

so

$$\begin{aligned} l'(\alpha) &= \frac{x_1 + x_6}{\alpha} + \frac{x_2 + x_3 + x_4 + x_5}{\beta} \cdot \frac{d\beta}{d\alpha} \\ &= \frac{x_1 + x_6}{\alpha} - \frac{x_2 + x_3 + x_4 + x_5}{2\beta} \end{aligned}$$

and

$$\begin{aligned} l''(\alpha) &= -\frac{x_1 + x_6}{\alpha^2} - \frac{x_2 + x_3 + x_4 + x_5}{\beta^2} \cdot \left(\frac{d\beta}{d\alpha}\right)^2 \\ &= -\frac{x_1 + x_6}{\alpha^2} - \frac{x_2 + x_3 + x_4 + x_5}{4\beta^2} \end{aligned}$$

so observed Fisher information is

$$J(\alpha) = \frac{x_1 + x_6}{\alpha^2} + \frac{x_2 + x_3 + x_4 + x_5}{4\beta^2}$$

and expected Fisher information is

$$\begin{aligned} I(\alpha) &= \frac{n\alpha + n\alpha}{\alpha^2} + \frac{n\beta + n\beta + n\beta + n\beta}{4\beta^2} \\ &= \frac{2n}{\alpha} + \frac{n}{\beta} \end{aligned}$$

Hence the Rao test statistic is

$$R = l'(\tilde{\alpha})^2 / I(\tilde{\alpha})$$

where $\tilde{\alpha}$ is the value specified by the null hypothesis (which is 1 / 6).

```
pi.zero <- pi.twiddle[1]
rao <- (((y[1] + y[6]) - sum(y[2:5]) / 2) / pi.zero)^2 / (3 * sum(y) / pi.zero)
print(rao)
```

```
## [1] 4.332
```

```
rao.pval <- pchisq(rao, 1, lower.tail = FALSE)
print(rao.pval)
```

```
## [1] 0.03740227
```

2.2.3.3 Wald Test

The Wald test depends on what we take to be the constraint function, say

$$g(\alpha) = \alpha - \tilde{\alpha}$$

(using the notation of the preceding section). Then the Wald test statistic is

$$W = (\hat{\alpha} - \tilde{\alpha})^2 I(\hat{\alpha})$$

```
alpha.hat <- pi.hat[1]
alpha.twiddle <- pi.twiddle[1]
beta.hat <- (1 - 2 * alpha.hat) / 4

wald <- (alpha.hat - alpha.twiddle)^2 * sum(y) * (2 / alpha.hat + 1 / beta.hat)
print(wald)
```

```
## [1] 4.254241
```

```
wald.pval <- pchisq(wald, 1, lower.tail = FALSE)
print(wald.pval)
```

```
## [1] 0.03915244
```

2.2.4 Summary

```
fred <- matrix(c(Gsq, rao, wald, Gsq.pval, rao.pval, wald.pval), ncol = 2)
rownames(fred) <- c("Wilks", "Rao", "Wald")
colnames(fred) <- c("statistic", "p-value")
round(fred, 4)
```

```
##       statistic p-value
## Wilks    4.3053  0.0380
## Rao      4.3320  0.0374
## Wald     4.2542  0.0392
```

IMHO the likelihood ratio test is easier here. But your choice. They all say approximately the same thing.

3 Two-Dimensional Tables

The data set read in by the R function `read.table` below

```
foo <- read.table(
    url("http://www.stat.umn.edu/geyer/5421/mydata/multi-simple-3.txt"),
    header = TRUE)
foo
```

```
##    y color    opinion
## 1 37   red       like
## 2 21  blue       like
## 3 25 green       like
## 4 37   red      so so
## 5 23  blue      so so
## 6 59 green      so so
## 7 49   red    dislike
## 8 29  blue    dislike
## 9 95 green    dislike
```

```
## 10 38   red no opinion
## 11 30  blue no opinion
## 12 57 green no opinion
```

has a count "response" variable and two categorical "predictor" variables that can be used for a regression-like formula. But these data can also be turned into a two-dimensional contingency table as follows.

```
yarray <- xtabs(y ~ color + opinion, data = foo)
print(yarray)
```

```
##        opinion
## color   dislike like no opinion so so
##   blue       29   21         30    23
##   green      95   25         57    59
##   red        49   37         38    37
```

We want to do the simplest hypothesis test for two-dimensional contingency tables, which is the only one covered in intro statistics books. It is called the test of independence (of the random variables whose values are the row and column labels) if both are random (that is we just collect individuals and the number in each cell is random with no constraints), or it is called the test of homogeneity of proportions if either the row margin or the column margin (but not both is fixed in advance).

In the jargon of categorical data analysis, the first (independence) is multinomial sampling, and the second (homogeneity) is product multinomial sampling. The latter name comes from the fact that if we fix (condition on) the row totals, then each row is a multinomial random vector that is independent of the others (with sample size that is the row total), and when things are independent probabilities multiply (hence "product"). We are also going to use a third sampling scheme, that the cells of the table are independent Poisson.

One goes from Poisson to multinomial by conditioning on the sum over all cells. One goes from multinomial to product multinomial by conditioning on the row totals (or the column totals, but not both). It turns out (this is a mysterious fact that will be cleared up in some other handout) that the MLE expected cell counts are the same for all three sampling schemes, hence the likelihood ratio test statistics, the Pearson chi-square statistics, and the degrees of freedom of their asymptotic chi-square distributions are *exactly the same* for all three sampling schemes. So it doesn't matter which one we use for calculations.

3.1 Likelihood Ratio Test

First use the Poisson sampling model and R functions `glm` and `anova`

```
out1 <- glm(y ~ color + opinion, family = poisson, data = foo)
out2 <- glm(y ~ color * opinion, family = poisson, data = foo)
aout <- anova(out1, out2, test = "Chisq")
print(aout)
```

```
## Analysis of Deviance Table
##
## Model 1: y ~ color + opinion
## Model 2: y ~ color * opinion
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1         6     15.623
## 2         0      0.000  6   15.623  0.01593 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
Gsq <- aout[["Deviance"]][2]
Gsq.pval <- aout[["Pr(>Chi)"]][2]
Gsq.df <- aout[["Df"]][2]
```

3.2 Pearson Chi-Square Test

Second use the multinomial sampling and R function `chisq.test`

```
cout <- chisq.test(yarray)
print(cout)
```

```
##
##  Pearson's Chi-squared test
##
## data:  yarray
## X-squared = 15.371, df = 6, p-value = 0.01756
```

```
names(cout)
```

```
## [1] "statistic" "parameter" "p.value"   "method"    "data.name" "observed"
## [7] "expected"  "residuals" "stdres"
```

```
Xsq <- cout$statistic
Xsq.pval <- cout$p.value
Xsq.df <- cout$parameter
```

3.3 Summary

```
fred <- matrix(c(Gsq, Xsq, Gsq.pval, Xsq.pval), ncol = 2)
rownames(fred) <- c("Wilks", "Pearson")
colnames(fred) <- c("statistic", "p-value")
round(fred, 4)
```

```
##         statistic p-value
## Wilks     15.6230  0.0159
## Pearson   15.3707  0.0176
```

Again, your choice.

3.4 Check

Let us check some of the assertions that these do exactly the same thing (asymptotically). First check that `anova.glm` and `chisq.test` agree on degrees of freedom.

```
Xsq.df == Gsq.df
```

```
##   df
## TRUE
```

Second check that `anova.glm` and `chisq.test` agree on expected values for all cells of the contingency table.

```
glm.expected <- predict(out1, type = "response")
glm.expected <- xtabs(glm.expected ~ color + opinion, data = foo)
all.equal(as.vector(glm.expected), as.vector(cout$expected))
```

```
## [1] TRUE
```

Third check that if we calculated the likelihood ratio test statistic from the result of `chisq.test` it would agree with what was produced by `anova.glm`.

```
all.equal(Gsq, 2 * sum(cout$observed * log(cout$observed / cout$expected)))
```

```
## [1] TRUE
```

Finally check that if we calculated the Pearson X^2 statistic from the result of `glm` it would agree with what was produced by `chisq.test`.

```
# redo glm.expected so it is in same order as in glm
glm.expected <- predict(out1, type = "response")
all.equal(as.numeric(Xsq), sum((foo$y - glm.expected)^2 / glm.expected))
```

```
## [1] TRUE
```

Everything checks.

The R functions `glm` and `chisq.test` use two different algorithms to fit the data, so the checks above would not agree exactly (would not agree if we replaced `all.equal` with `identical` in the checks above). But they do agree to within the inaccuracy of computer arithmetic when we make each use the same test statistic.