INF5830 – 2015, Solutions to set 4, exercise 3

Jan Tore Loenning

October 8, 2015

1 Comparing to a fixed reference value

We will take this step by step. First, consider the situation where it is well known that the best parser has a 90% accuracy. You are not testing the best parser. You are only testing your own parser. You test it on 1600 sentences and it parses 1460 of them correctly. You calculate this to be 91.25% accuracy and you wonder whether this is statistically significantly better than 90%.

Let us take one step back and reflect a little: why can't you just conclude that this is better than 90%? Think about flipping a fair coin. You expect it to come up heads half of the times you flip it. But you cannot from this conclude that it should always come up heads exactly 5 times when flipped 10 times. Choosing test sentences is similar to flipping a coin or—maybe a better analogy—drawing white and black balls from an urn. Suppose 90% of the balls are white and 10% are black. If you pick 1600 balls from the urn, you can't expect to draw exactly 1440 white balls. There might be more—there might be fewer. The question for us is: How unlikely is it that you draw 1460 or more white balls? Example sentences are like white and black balls. Whether you select a sentence that your parser can handle (=white) or one it can't handle(=black) is a random choice, just like drawing a ball.

So far the motivation. Let's move on to the test procedure. First we have to formulate the null hypothesis H_0 and the alternative hypothesis H_A.

- H_0 The accuracy of the parser ≤ 0.9.
- H_A: The accuracy of the parser > 0.9.

Observe that this is what is called a *one-sided* test: We are only testing whether the parser's accuracy is better than 0.9. If we instead wanted to test whether the parser's accuracy was different from 0.9—including the possibility that it was less than 0.9—we would have perfomed what we call a *two-sided* test. Observe also that H_A is not claiming that the parser's accuracy is 0.9125—only that it is better than 0.9.

To select one test sentence from all possible test sentences is what we call a Bernoulli trial. The outcome is 1 or 0. It is 1 with probability p, here

0.9. Hence to select 1600 sentences is the same as performing 1600 Bernoulli trials each with probability 0.9. (We here make the assumption that we are drawing from a larger population with a lot more than 1600 sentences. We are therefore entitled to think of it as a drawing with replacement.) Hence we can model this with the binomial distribution B(1600, 0.9). We want to answer how unlikely it is to get 1460 or more successes from this. This can be calculated as

- 1 - binom_cdf(1459, 1600, 0.9)

In principle, we could have used our own implementation of this function from oblig1. But it turns out the implementation is not able to handle such large numbers. Hence we turn to the SciPy implementation and calculate

- 1 - stats.binom.cdf(1459, 1600, 0.9)

The answer is 0.05003 (if we include 5 decimal places). Hence, it is exactly on the border for whether we will conclude that the test is statistically significant at the level 0.05. Strictly speaking it is not.

Exercise See how large a difference one test result can make by calculating

- 1 - stats.binom.cdf(1460, 1600, 0.9)

Why isn't this the correct calculation given our observation?

1.1 Approximation by a normal distribution

Nowadays we can calculate the binomial distribution itself given proper software. But in the older days one had to use normal distributions to approximate the binomial distribution. It may still be useful in many situations to use the normal distribution approximation. We will therefore also consider how the same task can be solved by using the approximate normal distribution.

To determine which normal distribution that approximates a particular binomial distribution, one needs to calculate the mean and the standard deviation. The mean is $\mu = n \times p = 1440$. For the standard deviation, we first consider the variance. We know that the variance of one Bernoulli trial is $p(1-p) = 0.9(1-0.9) = 0.09$. And since we consider n independent trials, the variance of n trials is $np(1-p) = 1600 \times 0.9(1-0.9) = 144$. The standard deviation is $\sigma = \sqrt{np(1-p)} = \sqrt{1600 \times 0.9(1-0.9)} = \sqrt{144} = 12$. Hence we use the standard distribution

$$N(\mu, \sigma) = N(np, \sqrt{np(1-p)}) = N(1440, 12)$$

We calculate the z-score from the observation:

$$z = \frac{\overline{X}_B - \mu}{\sigma} = \frac{1460 - 1440}{12} = \frac{5}{3}$$

To find the corresponding p-value, we may use a table or the normal distribution in SciPy.

- 1 - stats.norm.cdf(5.0/3)

This yields 0.04779 (with five decimal places). This is quite close to the number we found using the binomial distribution. In particular, it is between the values we get for the binomial distribution for 1459 and 1460. But interestingly enough, it is sufficiently different from the number we get by using the cdf of the binomial distribution (with 1459) that this time we could have concluded that the null hypothesis is refuted at the $p = 0.05$-level.

What can we learn from this approximation? Maybe that we should not conclude too much from the last decimals in comparing p to 0.05. The number 0.05 itself is also chosen somewhat arbitrarily.

1.1.1 Proportions

We could have stated the same problem by instead talking of proportions. The proportion of successes is $\hat{p} = \frac{\text{count of successes}}{\text{sample size}} = \frac{1460}{1600} = 0.9125$. We can then ask how unlikely it is to get this proportion when we expect 0.9. For the normal approximation we now use the distribution:

$$N(\mu, \sigma) = N(p, \sqrt{\frac{p(1-p))}{n}} = N(0.9, 0.075)$$

We calculate the z-score from the observation:

$$z = \frac{\hat{p} - p}{\sigma} = \frac{0.9125 - 0.9}{0.0075} = \frac{5}{3}$$

We see that we get the same result as when we used normal approximation to the binomial distribution.

2 Calculating a confidence interval

Let us now restate the problem. Say we have no reference parser to compare to. All we have are the results from testing our own parser, i.e., 1460 successful parses out of 1600 test items. This gives a success rate of 0.9125. We want to estimate the true accuracy with a confidence level of .95.

We will use the normal distribution approximation for this. First we have to establish a $z*$ such that the area C of the standard density curve between $-z*$ and $z*$ is 0.95. Observe how this differs from the testing above. It corresponds to a two-sided test. We use probability mass 0.025 below $-z*$ and the same above $z*$.

We know that the corresponding value is $z* = 1.96$. (If you don't remember this number, you may use a table or software. In SciPy the command is stats.norm.ppf(0.975)).

We do not know the true standard deviation. We estimate it from our sample and calculate the sample standard deviation (sometimes called the standard error).

$$\mathrm{SE}_{\hat{p}} = \sqrt{\frac{\hat{p}(1-\hat{p})}{n}} = \sqrt{\frac{0.9125(1-0.9125)}{1600}} = 0.00706416$$

This yields the interval

$$[\hat{p} - z^* \mathrm{SE}_{\hat{p}}, \hat{p} + z^* \mathrm{SE}_{\hat{p}}] = [0.8986545, 0.9263455]$$

3 Comparing two sample proportions

Let us now turn to the situation described in the exercise. We have two parsers. We test each of them on 1600 sentences. Parser A parses 1440 of them correctly. Parser B parses 1460 of them correctly. Can we conclude that parser B is better than parser A? We formulate the null hypothesis and the alternative hypothesis.

- H_0: Parser B's accuracy $\leq$ parser A's accuracy.
- H_A:Parser B's accuracy $>$ parser A's accuracy.

A general method for comparing two different samples and see whether they can come from the same population is the two-sample t-test. It can be used to compare the height of say Norwegian and Swedish men. We may select M Norwegian men and N Swedish men and calculate the t-score

$$t = \frac{\hat{x}_{No} - \hat{x}_{Sw}}{\sqrt{\frac{s^2_{No}}{M} + \frac{s^2_{Sw}}{N}}}$$

and then use a t-distribution to find the corresponding p-value. Observe that M and N may be different. This method was used in exercise 1 of this exercise set for comparing the sentence length across different genres.

(The rules for selecting which $t(k)$-distribution to use are not so easy to get. More and McCabe e.g., recommends to use software or the smaller of $M-1$ and $N-1$ for k, which is a conservative lower bound. The built in test in SciPy, called *stats.ttest_ind*, seems to use $M+N-2$ for k.)

Anyway, that is of less importance for us here since we will test for proportions. The test then gets the form

$$z = \frac{\hat{p}_B - \hat{p}_A}{\mathrm{SE}_D}$$

where we for SE_D may use

$$\mathrm{SE}_D = \sqrt{\frac{\hat{p}_A(1-\hat{p}_A)}{n_A} + \frac{\hat{p}_A(1-\hat{p}_B)}{n_B}}$$

(There are alternative ways to estimate SE_D which in some situations give more accurate approximations, but that is not important here for our modest purposes).

Working with proportions it is usual to use the normal distribution itself, also called the z-distribution, and not any t-distribution. (Observe by the way that for large n the t-distribution comes close to the z-distribution.)

Plugging inn $\hat{p}_A = 0.9125$, $\hat{p}_B = 0.9$, and $n_A = n_B = 1600$ we get a z-score of 1.2132358 and a p-value of 0.1125. Hence we cannot reject the null hypothesis.

We could here have chosen to use the built in t-test of SciPy directly.

- A = [1 for i in range(1440)]+[0 for i in range(160)]
- B = [1 for i in range(1460)]+[0 for i in range(140)]
- stats.ttest_ind(A,B)
- which returns (-1.212856..., 0.22527...)

The first returned number is the t-score, the second number is the p-value. The p-value is here for the two-sided test. As we see, the numbers aren't very different from what we get when using the z-distribution.

4 Pairwise comparison

Let us first consider an analogous example. Say we are to test out a new drug for lowering the cholesterol level in the patients' blood. We let a sample of persons with high cholesterol take the drug for three months. We will measure the cholesterol level at the beginning of the experiment and after three months and compare. There are several ways one could proceed.

One possibility is to measure the cholesterol level of all the patients at the beginning of the experiment and calculate the mean. And then do the same after three months: find the mean of the cholesterol level of all the patients. One can then measure the difference between the two means and see whether the latter mean is significantly lower than the first mean. One possible outcome is that the second mean is lower, but that the difference is not statistically significant thanks to the large variance within the group.

Another approach would be to for each patient to measure the cholesterol level at the beginning of the experiment and then again after three months and take the difference. One may then consider the sample of the differences for all the patients, take the mean and check whether that is significantly below zero. We might get a statistically significant result with the second procedure even if we don't get it by the first procedure.

Also when comparing two classifiers, we may take two similarly different approaches. The first is what we did in the last section, evaluating each of

the two classifiers on its own test set, taking the average and then compare to the results for the other classifier. The other approach is to test the two classifiers on the same test items and compare them item for item. If one of the classifiers is correct on all items where the other is correct and on a few more items, we might conclude that it is a better classifier, even though the difference in measured accuracy is not large.

Let us see how the given example is doing. If we use 1 for success and 0 for failure, the variable X_C will take the value 1 for 45 items in the sample, the value -1 for 25 items, and the value 0 for the remaining 1530 items.

We then have to formulate our hypotheses

- H_0: $\overline{X}_C \leq 0$
- H_A: $\overline{X}_C > 0$

before we can carry out a one-sided the test.

When I did that I got a t-value of 2.3940 and a p-value of 0.00839. We can then refute the null hypothesis not only on the 0.05 level, but even on the 0.01 level.

4.1 The sign test

There is one problem with using the t-test, t-distribution or z-distribution like this. They all assume a normal distribution of the data to be correct. If the data is not normally distributed, these distributions are only more or less accurate approximations. Tests which assume some particular distributions are called *parametric*.

We cannot assume the variable X_C to be normally or binomially distributed. It is considered more correct to use a so-called *non-parametric* test which makes no such assumptions. The simplest such test is the *sign test*. This test only considers the items where the two classifiers yield different results. It disregards all items where the two agree. It then counts how many times one is better and how many times the other is better, here 45 and 25. The expectation is that if the two classifiers are equally good, these numbers should be roughly equal. We therefore ask how likely it is that parser B performs better than parser A on 45 or more out of 70 items? A way to measure this is to use the binomial distribution for 70 items with p=0.5 and ask how probable it is to get 45 or more successes. We can calculate this using SciPy as follows.

- 1 - stats.binom(44, 70, 0.5)

The answer is 0.01123. The number is a little bigger than the one we got using the t-test, but it more than suffices to conclude that parser B is better than parser A at the 0.05 level.

4.2 Bootstrapping

Jurafsky and Martin, 3.ed, sec. 7.3 describes an alternative way for carrying out nonparametric tests where they apply bootstrapping. Unfortunately, the description in the book is to dense to understand what they are doing. To explain it in an understandable way would take 3–5 pages. Therefore, we forget about their procedure for the moment. We might return to it towards the end of the semester, but don't worry about it for the time being.