Code-testing of Statistical Test implementations

F. James, A. Pfeiffer, A. Ribon
CERN
P. Cirrone, S. Donadio, S. Guatelli, A. Mantero, B. Mascialino, L. Pandola, S. Parlati, M.G. Pia
INFN
P. Viarengo
IST

In this note we discuss in general how to test the implementation code of statistical tests, and then we treat in detail the case of the Kolmogorov-Smirnov test. It will be shown that some "obvious" expected properties, like the flatness distributions of p-values from repeating drawings from the same parent distribution, are not indeed reproduced even in absence of bugs in the code, due to either asymptotic approximations in the formulas used to compute the p-value, or to the discreteness of the distance distribution in the case of direct Monte Carlo evaluation of the p-value. This makes the code-testing more complicated. Some practical advice is presented anyhow.

1. Introduction

It is essential, before using any statistical test (χ^2, Kolmogorov-Smirnov, Cramer-von Mises, Anderson-Darling, etc.), to check whether its code implementation is correct. The obvious way to do so is to compare the results in some particular cases against either different implementations of the same statistical test, or against some formulas or tabulated values from the Statistics literature. This approach is quite limited (for example, to the best of our knowledge, there are no publicly available implementations of the Anderson-Darling statistical test), and even in the few situations when a formula or a table (for the p-value given the distance) is available, it is usually obtained under some assumptions, the most typical one being the asymptotic limit of the sample size, and it is difficult, in general, to know what is the bias on the p-value caused by such approximations. We present here a method which does not rely on external code, papers, books, or tables, to validate any implementation of any statistical test. The method is based on the expected mathematical properties that any statistical test should exhibit, which are checked using Monte Carlo trials. Although we believe that the method provides a reasonable and powerful way to detect bugs in code implementations, it cannot give an absolute guarantee that the code is completely bug-free.
We will consider uniquely 1-dimensional distributions.

2. Testing strategies

We aim here to be quite general, so we will use some symbolic notation and our discussion will be somehow abstract, but we will be soon back to a concrete example in the next section. For the same reason, in the following we will not say anything on the continuous or discrete nature of the parent distribution and whether the sample data should be binned or unbinned.

Let S_1 and S_2 be two 1-dimensional samples of size N_1 and N_2 respectively, and $d(S_1, S_2)$ be a test statistic measuring the distance between the two samples. We can then calulate $T(S_1, S_2)$, the probability that $d(S_i, S_j)$ would not be smaller than $d(S_1, S_2)$, for any samples S_i and S_j of size N_1 and N_2 drawn randomly from the same parent distribution. T is called the *p-value*. Here are some simple properties of T:

i) $T(S_1, S_2) = 0$
when the two samples are in non-overlapping regions of the real axis.

ii) $T(S_1, S_1) = 1$
i.e. when the two samples are identical.

iii) $< T(S_1', S_2') > \ \geq \ < T(S_1'', S_2'') > \ \geq \ \ ...$
where $< T >$ means the average of the p-values obtained from some drawings from the same parent distribution, for both samples S_1 and S_2, but with increasing shifts. For example, let's consider as S_1' and S_2' the samples drawn from the same gaussian distribution $N(\mu, \sigma)$; then consider as S_1'' and S_2'' the samples drawn respectively from $N(\mu + \sigma, \sigma)$ and $N(\mu - \sigma, \sigma)$; and so on, for less and less overlapping gaussian parent distributions.

iv) $T(S_1, S_2) = T(f(S_1), f(S_2))$
for any monotonic function $f(x)$. Notice that this property is not rigorously valid in the case of binned distributions.

v) $T(S_1, S_2) = T(S_2, S_1)$
i.e. the test should not depend on the order of the two samples, that is on which one we label as "1" and "2".

vi) The above properties should be valid independently of the parent distribution from which we draw the samples. In practice, we think that

some reasonable choices for the parent distribution can be the following: flat (uniform), gaussian, left-tailed and right-tailed exponential.

Suppose that the statistical test fulfills all the above requirements; then we can move to the next step, which is much more CPU demanding and trickier.

We define as *Pseudoexperiment* a random drawing of two samples, S_1 of size N_1 and S_2 of size N_2, from the same parent distribution (whatever it is). Given these two samples, we can calculate the *distance*, d, between them according to the statistical test we are considering, and then from that distance we can calculate the corresponding p-value, p. For each pseudoexperiment we thus have: $S_1^{(j)}, S_2^{(j)} \rightarrow d^{(j)}$, $p^{(j)}$ where $j = 1, 2, ..., N$, with N number of pseudoexperiments. Now, from the distribution of distances, $d^{(j)}$, $j = 1, 2, ..., N$ we can calculate the p-value directly from its definition: *the p-value of a given distance $\overline{d}$ between two samples S_1 and S_2 (with respect to a given statistical test) is the probability to get a distance $d \geq \overline{d}$ between two samples of the same size as S_1 and S_2 drawn from the same parent distribution (whatever it is).* In practice, the above probability is estimated as *the fraction of pseudoexperiments whose distance $d^{(j)} \geq \overline{d}$*. We call this operative definition of the p-value *Monte Carlo p-value*, p_{MC}.
Notice that it is important to include the *equal* case in $d \geq \overline{d}$: this of course would not matter for real continuous distributions, but in practice we are always dealing with discrete distributions of distances.
Concretely, one can consider either $N_1 = N_2$ or $N_2 >> N_1$; in the latter case, one could think also to draw the second sample, the one with higher statistics, only once instead of for each pseudoexperiment; as limiting case of $N_2 \rightarrow \infty$, one can make a 1-sample statistical test, comparing directly S_1 with the parent distribution, at least in the cases in which the analytic expression of its cumulant probability distribution is known. We will compare these possibilities in the example of the next section.

Naively, we would be tempted to require as necessary properties of the statistical test under consideration the following two:

a) In the limit of a large number of pseudoexperiments, N, the distribution of p-values (obtained from the statistical test), $p^{(j)}$, should be a flat (uniform) distribution between 0 and 1, hence, in particular, it should have: $\mu = \frac{1}{2}$, $\sigma = \frac{1}{\sqrt{12}}$ (where μ is the mean, and σ the rms).

b) Apart for tiny deviations due to finite numerical accuracy, the p-values determined from the statistical test have to coincide with the ones determined directly from Monte Carlo: $p^{(j)} = p_{MC}^{(j)}$.

Both properties are *not* true, for two independent reasons. The first one is due to the fact that p-value computed by the statistical test is usually valid under some "asymptotic" conditions, the most general one being the limit of the sample size (in our case above, N_1 and N_2) to ∞. The second one, more subtle, is due to the *discreteness of the distance distribution* $\{d_1, d_2, ...\}$, even in the limit of a very large number of pseudoexperiments, $N \rightarrow \infty$. As as consequence of this, even the following property, which is the analogous of a) for p_{MC}, does *not* hold:

a') In the limit of a large number of pseudoexperiments, N, the distribution of p-values calculated directly from Monte Carlo, $p_{MC}^{(j)}$, should be a flat (uniform) distribution between 0 and 1, hence, in particular, it should have: $\mu = \frac{1}{2}$, $\sigma = \frac{1}{\sqrt{12}}$.

It is even possible to find a very simple formula which predicts the mean value of $\left\{p_{MC}^{(j)} \; ; \; j = 1, 2, ..., N\right\}$, given the multiplicities of the various distances, i.e. the number of times that each different distance appears:
distance d_1 with multiplicity M_1 ;
...
distance d_K with multiplicity M_K ;
where K is the number of *different* distances, and $\sum_{i=1}^{K} M_i = N$:

$$< p_{MC} >= \frac{1}{2} + \frac{1}{2N} + \sum_{i=1}^{K} \frac{M_i \left(M_i - 1\right)}{2N^2} \qquad (1)$$

Notice that:

- $< p_{MC} > \; > \; \frac{1}{2}$ in all cases (with finite N);
- for a given N, the lowest value of $< p_{MC} >$, that is the closest to $\frac{1}{2}$, is reached when $K = N$, that is when all the distances are different: $M_1 = 1$, ... , $M_K = 1$: $< p_{MC} >= \frac{1}{2} + \frac{1}{2N}$
- in order to get $< p_{MC} > \rightarrow \frac{1}{2}$ not only $N \rightarrow \infty$ is necessary, but also $K/N \rightarrow 1$, i.e. only a finite number of distances can be repeated;
- $< p_{MC} >$ depends only on the multiplicities of the K different distances, but not on the explicit values of these distances.

The consequence of the above facts is that the task of checking the code implementation of a statistical test becomes much harder, because, for instance, discrepancies between computed p-values and direct Monte Carlo ones are expected even with no bugs in the code. However, these should decrease as the asymptotic conditions are approached. Notice that, in the case the remaining discrepancies are judged unacceptable but the implementation of the p-value is correct and a better formula for the p-value cannot be found, the direct Monte Carlo p-value can always be employed. The only drawback of this approach is that it is quite CPU intensive.

3. An example: the Kolmogorov-Smirnov test.

For the Kolmorogov-Smirnov test we use the p-value formula given in [1]. As parent distribution we consider the flat (uniform) distribution between 0 and 1, because in this case the cumulant probability distribution is known ($F(x) = x$). $N = 100\,000$ pseudoexperiments have been generated, of four different types as defined by the way the distance has been determined:

d1 : in each pseudoexperiment we draw a single sample S_1 of size N_1, and then we consider the 1-sample Kolmogorov-Smirnov test against the parent distribution $F(x) = x$.
Hereafter we indicate with **d1** the corresponding distance.

d2a : in each pseudoexperiment we draw a single sample S_1 of size N_1, and then we consider the 2-sample Kolmogorov-Smirnov test against another sample, S_2, of very large size, $N_2 = 10\,000$, which is drawn from the same parent distribution, but only once (at initialization, not in each pseudoexperiment).
Hereafter we indicate with **d2a** the corresponding distance.

d2b : in each pseudoexperiment we draw two samples, S_1 of size N_1, and S_2 of very large size $N_2 = 10\,000$, and then we consider the 2-sample Kolmogorov-Smirnov test between them.
Hereafter we indicate with **d2b** the corresponding distance.

d2c : in each pseudoexperiment we draw two samples, S_1 and S_2, of the same size N_1, and then we consider the 2-sample Kolmogorov-Smirnov test between them.
Hereafter we indicate with **d2c** the corresponding distance.

As sample size N_1 we consider the following possibilities: 10 , 50 , 100 , 500 , 1000 , 5000 , $10\,000$. For the sample size N_2, when not equal to N_1, we use $N_2 = 10\,000$. Only for the case $N_1 = 1000$, to see what happens when N_2 is changed, we also consider $N_2 = 100\,000$, i.e. an increase of a factor ten.
For each of the above four types of distances (and, of course, for each pseudoexperiment) we determine two types of p-value: p, the analytic p-value, and p_{MC}, the p-value from the direct Monte Carlo method. The table on the right side reports the summary of our study. In the first column there is N_1, the size of the sample S_1; in the second column there is the type of distance; in the third column there is the number of different distances (i.e. what we have called "K" in the previous section); in the last two columns there are the mean values (over the N values obtained in the pseudoexperiments) of the two different types of p-values, p, and p_{MC} (in the latter case, such mean value agrees with the one predicted by (1)).

N1	Type	distances	$< p >$	$< p_{MC} >$
10	d1	99,768	0.6605	0.5000
	d2a	4,421	0.5118	0.5002
	d2b	4,467	0.5120	0.5002
	d2c	10	0.5509	0.6290
50	d1	99,906	0.5863	0.5000
	d2a	2,241	0.5103	0.5004
	d2b	2,261	0.5097	0.5004
	d2c	24	0.5304	0.5585
100	d1	99,864	0.5637	0.5000
	d2a	1,651	0.5101	0.5006
	d2b	1,654	0.5085	0.5006
	d2c	31	0.5237	0.5415
500	d1	99,712	0.5286	0.5000
	d2a	776	0.5129	0.5013
	d2b	802	0.5030	0.5013
	d2c	67	0.5112	0.5186
1,000	d1	99,547	0.5210	0.5000
	d2a	569	0.5185	0.5019
	d2b	595	0.5034	0.5018
	d2c	92	0.5076	0.5132
x10 S2 size	d2a	4,643	0.5028	0.5002
	d2b	4,646	0.5026	0.5002
5,000	d1	99,131	0.5087	0.5000
	d2a	288	0.5722	0.5039
	d2b	335	0.5020	0.5034
	d2c	193	0.5035	0.5059
10,000	d1	98,827	0.5056	0.5000
	d2a	208	0.6441	0.5056
	d2b	278	0.5027	0.5041
	d2c	268	0.5022	0.5042

From the table on the right side we can make the following observations:

- The number of different distances grows with:
  a) the number of pseudoexperiments;
  b) the sample size, in the case of two samples of equal size drawn in each pseudoexperiment;
  c) inversely with the sample size of the first sample, in the case that the second sample has a much larger size, and no matter whether is drawn once or each time;
  d) the sample size of the second sample, in the case the latter is much bigger than the first one, and no matter whether is drawn once or each time.

In the case of a single sample, that is when we compare the sample directly with the parent distribution, the number of different distances is almost always equal to the number of pseudoexperiments. In the case of two samples, but with the second of much higher size, the number of different distances is always slightly bigger (but very little) in the case of drawing of both samples in each pseudoexperiment, with respect to the case of a single drawing for the second sample.

- The mean of the Monte Carlo p-values, p_{MC}, depends only on the number of different distances, and their multiplicities, as predicted from (1);
- The means of the theoretically calculated p-values, p, have the following characteristics:
  a) they are systematically above $\frac{1}{2}$;
  b) for d1, d2b and d2c, p gives mean p-values which are closer to $\frac{1}{2}$ the larger the sample size N_1 is; however, in the case of d2b, a "saturation" sample size is reached for values around $N_1 = 500$;
  c) for d2a, p gives mean p-values which are not always getting closer to $\frac{1}{2}$ the larger the sample size gets, because N_1 gets closer to N_2 but we draw the second sample only once.

4. Conclusions

From the study we have presented it is possible to draw some useful practical suggestions on code-testing of statistical test implementations. Although we treated here explicitly only the case of the Kolmogorov-Smirnov test, we believe that such advices are valid in general, for any statistical test.

1) First of all, start by checking the properties i) ÷ vi) (see the section "Testing strategies"), and move on only when they are all satisfied.

2) Generate a very large number N of pseudoexperiments (e.g. 100 000), and for each pseudoexperiment do the following:
draw a sample S_1 of size N_1 (a fixed, arbitrary value, e.g. $N_1 = 100$), and a sample S_2 of size $N_2 >> N_1$ (e.g. $N_2 = 10\,000$), from the same parent distribution (whatever it is), and then calculate the distance and the p-value of the statistical test under consideration.
(Notice that a 2-sample, rather than 1-sample, statistical test is used because in general it is not possible to find an analytical expression for the cumulant probability distribution of a given parent distribution.)

3) From the distribution of distances, calculate the direct Monte Carlo p-value for each distance (i.e. pseudoexperiment).

4) Calculate the average of the direct Monte Carlo p-values: this has to coincide exactly with what is predicted by (1) (to apply this formula only the multiplicities of the different distances are needed). If this is not the case, then there is something wrong in the testing code itself (don't blame the statistical test implementation). Move on only when the two agree.

5) Calculate the average of the p-values returned by the statistical test, and the average and maximum absolute difference between these p-values and the corresponding direct Monte Carlo ones.

6) Repeat 2) ÷ 5) for few different values of N_1 (e.g. $N_1 = 100, 500, 1000, 5000$). You should observe a convergence, as N_1 grows, between the p-values returned by the statistical test and the direct Monte Carlo ones. If this is not the case, then there is something wrong in the statistical test implementation either with the distance calculation or with the p-value determination.
Finally, if such convergence is indeed observed, one should judge whether the average and maximum absolute difference of the p-values returned by the statistical test and by the direct Monte Carlo method, in the case of the highest N_1 value (e.g. 5000), look "reasonable" under the assumption that they are entirely due to the asymptotic approximations on which the p-value formula (or table) is based on. If this is not the case, then the implementation of such p-value should be first checked, and if it is fine, then a better formula should be used instead (if it can't be found, the direct Monte Carlo p-value can be employed; eventually, if it is too slow to do on the fly, one could store the Monte Carlo results on a table once for all).

References

[1] W.H Press, S.A. Teukolsky, W.T. Vetterling, B.P. Flannery *Numerical Recipes in C*, Cambridge (see chapter 14).