Test for Randomness

Jiashu Xu

November 2020

1 Introduction

In this essay I would investigate on the randomness of the shuffle. How to test for randomness?

There are two subsections and each covers one specific type of shuffle, i.e. shuffle the data with repeated items or shuffle the data with each item being unique.

Assume we have the data $X = \{x_1, ..., x_n\}$, and we want to shuffle X to produce a permutation.

First I would present an efficient way to generate shuffle [1].

Algorithm 1 Fisher-Yates

```
    Input: $X = \{x_1, ..., x_n\}$
for i = 0, ..., n-2 do
    $j \sim DisUni(i, n-1)$                ▷ j random integer from i to n-1
    swap $x_i$ and $x_j$
```

Theorem 1.1. *Fisher-Yates generates unbiased shuffle.*

Proof. A shuffle is unbiased iff each permutation is equal-likely.

Consider one permutation X' of X, let $X' = \{a_1, ..., a_n\}$.

$$\begin{aligned} P(\text{Fisher-Yates produces } X' \text{ from } X) &= P(\text{each } X'_i = a_i) \\ &= P(X'_1 = a_1) \cdot P(X'_2 = a_2 \mid X'_1 = a_1) \cdot ... \\ &P(X'_n = a_n \mid X'_1 = a_1, ..., X'_{n-1} = a_{n-1}) \\ &= \frac{1}{n} \cdot \overbrace{\frac{1}{n-1}}^{\text{j from 1 to n-1}} \cdot ... \cdot 1 \\ &= \frac{1}{n!} \end{aligned}$$

[1] I used 0-based notation

□

In this essay I would use Fisher-Yates to generate shuffle. But I would assume that I do not know that this algorithm would generate the unbiased shuffle, and would construct statistical test to test for randomness. Moreover, in section 3 I would investigate some methods for testing randomness, since shuffle is also one way to generate randomness.

2 Permutation of unique items

Assume X is a permutation of 123...n.

2.1 χ^2 test / Frequency test

We can calculate the number of times each items happen in one specific location, say, the first item.

For $1 \leq i \leq n$,

$$P(X_1 = i) = \frac{1}{n}$$

So can use χ^2 test. However, note that it can test the probability of each item shows up in one specific position, but can't test the correlation between the items, so it is weaker.

3 Permutation of duplicate items

For this setting we let sequence $X = \{x_1, ..., x_n\}$, assume that each $X_i \in \{0, ..., d-1\}$. Note since each x_i is assumed to be independent, this sequence can have duplicated items.

3.1 χ^2 test / Frequency test

Similar to 2.1 we can also use χ^2 test.

3.2 poker test / Partition test

This test[2] is a modified χ^2 test. For the sequence X, and we can group the n items into blocks of k items. For each individual k items we can check how many different values are there.

[2] Knuth The Art of Computer Programming chapter 3.3.2 D

For $r = 1, ..., k$,

$$P_r = P(\text{k items have r different values})$$

$$= \frac{\overbrace{d \cdot ... \cdot (d-r+1)}^{\text{1st item d possible, 2nd d-1...}}}{\underbrace{d^k}_{\text{k items, each d possible}}} \cdot \overbrace{\begin{Bmatrix} r \\ i \end{Bmatrix}}^{\text{number of ways to partition k items into r parts}}$$

in which $\left\{ {r \atop i} \right\}$ is the Stirling number of 2nd kind (see Probability [6].1.3°).

Now we can run the test on the input sequence X: split X into $\lfloor \frac{n}{k} \rfloor$ blocks, classify each block into k category, and increase the counter for that category. Once all $\lfloor \frac{n}{k} \rfloor$ blocks are examined we have $T_1, ..., T_k$, each represent the total number of blocks of k items that have i different values; and $P_1, ..., P_k$ that are calculated from the equation above.

For example we can test on Random Number Generator with H_0: each digit is random. Since each digit is from 0 to 9. $d = 10$, if we set $k = 3$, and consider the X to be the 1,000,000 digits number generated by RNG, then one sample run might be

Figure 1: One sample run

differences	observations	theoretical probability
1 (All same)	3312	0.01
2	89547	0.27
3 (All different)	240474	0.72

In fact, as some papers claim[3], in reality it is always better to manually tune the k value to achieve better confidence. A suggested method is to do poker test on $k = 3, 4, 5$.

3.3 serial test

Also[4] a modified χ^2 test, consider a sequence X with n even. We can split X into blocks $B_1, ..., B_n$ with each block having 2 items B_{i1}, B_{i2}. Since each $X_i \in \{0, ..., d-1\}$, we have $0 \leq B_{i1}, B_{i2} \leq d-1$. For a particular pair of integer

[3] Testing Randomness: Poker Test with Hands of Three Numbers

[4] Knuth The Art of Computer Programming chapter 3.3.2 B

(p_1, p_2) that $0 \le p_1, p_2 \le d-1$, the probability $P\big((B_{i1}, B_{i2}) = (p_1, p_2)\big) = \dfrac{1}{d^2}$, and we can do χ^2 test on those $k = d^2$ pairs.

Also note that for a valid χ^2 test we need n somewhat large compared to k, like $2n \le 5k = 5d^2$.

3.4 gap test

Also[5] a modified χ^2 test. We pick $0 \le k \le d-1$, and for this digit K, calculate the length of gap between one k and another k, Also define the first gap is the gap between the first occurrence of k and the second occurrence of k; and the last gap is the 2nd to last occurrence of K to the last occurrence of k.

For example, in binary string 10010011010, the gap for $k = 1$ is 2,2,0,1. So we have

length of gap	counter
0	1
1	1
2	2

If there is a gap of length 10, then $P(\text{gap of }10) = P(\text{not k})^{10} \cdot P(k)$, for example if d = 10 and k =3, $P(\text{gap of }10) = (0.9)^{10} \cdot 0.1$.

Consider the cdf $F(x) = \frac{1}{d} \cdot \sum_{i=0}^{x} (\frac{d-1}{d})^i = 1 - (\frac{d-1}{d})^{x+1}$.

The gap test is the following:

Algorithm 2 Gap Test

```
    Input: $X = \{x_1, ..., x_n\}$, each $X_i \in \{1, ..., d-1\}$
Let GAP be a list
for k = 0, ..., d-1 do
    calculate gaps on $k$, and append to GAP
Run Kolmogorov-Smirnov test on GAP with cdf $F(x) = 1 - (\frac{d-1}{d})^{x+1}$
```

3.5 auto-correlation test

Consider random sequence $X_1, ..., X_n$ is sampled in time $t_1, ..., t_n$, the lag-k auto-correlation is

$$r_k = \frac{\sum_{i=1}^{n-k} (X_i - \bar{X}_i)(X_{i+k} - \bar{X})}{\sum_{i=1}^{n} (X_i - \bar{X})^2}$$

although we do not use t_i in r_k equation, we assume that each t_i is equally-spaced.

[5] Knuth The Art of Computer Programming chapter 3.3.2 D

We generally set $k = 1$ when testing for randomness [6].

3.6 Wald-Wolfowitz / run test

This test is restricted to d = 1, i.e. binary sequence input.

Assume that $X = \{X_1, ..., X_N\}$ with $m+n = N$ has n 1s and m 0s. Then the number of runs R in X has $\mu = \mathbb{E}[R] = 1 + \frac{2nm}{n+m}$ (see Probability $\boxed{16}$.eg.5); and $\sigma^2 = \frac{2nm(2mn - N)}{N^2(N-1)} = \frac{(\mu-1)(\mu-2)}{N-1}$.

In Annals Math. Stat. 15 (1944), 163-165 Wolfowitz proved that when N large $R \xrightarrow{D} N(\mu, \sigma^2)$, thus we can use test statistic $Z = \frac{R-\mu}{\sqrt{\sigma^2}}$.

We can potentially extend run test to accept input of $d \geq 3$ by defining run to be a succession of similar events proceeded and followed by a different event. For example, we can define event to be increasing sequence or decreasing sequence.

[6] Beside testing if sample data is generated from a random process, it can also test whether to use non-linear or time series model to model these data