Applied Poker Test for General Digital Sequences

Sahar A. Mohammed

Abstract: The Poker test is considered one of the important statistical randomness tests. This test, for independence, is based on the frequency in which certain digits are repeated in a series of numbers. In this paper we will attempt to generalize the binary poker test to be suitable to applied on not only binary sequences, in another word, the generalized poker test could be applied on digital (m-) sequences (for m≥2). The generalized poker test called Grouped poker test.

Keywords: Statistical Randomness Tests, Poker Test, Hypothesis Test, Chi-Square Test, cryptography.

I. Introduction

The problem of testing randomness is motivated by the need to evaluate the quality of different random number generators used by many practical applications including computer simulations, cryptography and communications industry [1].

The poker test treats numbers grouped together as a poker hand. Then the hands obtained are compared to what is expected using the chi-square test ([2]).

Definition(1) [3]: A Pseudo Random Bit Generator (PRBG) is a deterministic algorithm which, given a truly random binary sequence of length k, outputs a binary sequence of length L, k which “appears” to be random. The input to the PRBG is called the seed, while the output of the PRBG is called a pseudorandom bit sequence.

Remark (1) [3]: The χ² (chi-square) distribution can be used to compare the goodness-of-fit of the observed frequencies of events to their expected frequencies under a hypothesized distribution. The χ² distribution with ν degrees of freedom arises in practice when the squares of ν independent random variables having standard normal distributions are summed.

In 1982, the five statistical tests for local randomness are being found to be applied on binary key stream sequences [4]. In 1994, a new package of randomness introduce by CRYPT-X [5]. This package is a microcomputer package that is intended to be used to test either large binary strings that are to be used as key stream in stream ciphers or block cipher algorithms. One of the tests of this package is subblock test which is a similar to poker test, its partitions the stream into F hands of length m bits. For a stream of size n, where F/2^m≥5, the total number of hands is ⌊n/m⌋, where ⌊ ⌋ denotes the integer value. The aim of this test is to show that there is an equal number of each of 2^n possible hands. If f_i denotes the frequency of hand pattern s_i, then the test statistic used is [6]:

\[ T=2^{-n} \sum_{i=0}^{2^m-1} f_i^2 - F \]  

...(1)

This compared with chi-squared distribution with degree of freedom equal to 2^n-1.

In 2009, Ali et al. [3] introduce a paper that include the generalization of three basic and important tests of the standard statistical basic randomness tests, these tests are frequency, Run and Autocorrelation tests.

In 2014, Ibraheem [7], extends the 2-tuple to d-tuple (d≥3) to apply binary serial test for binary sequences. Second, she generalized the 2-tuple binary serial test to 2-tuple digital serial test for digital (m-) sequences (m≥3) generated from digital generators. Lastly, she extends the 2-tuple to d-tuple digital serial test for digital sequences.

The results of randomness tests of applying the binary and digital poker tests in the binary and digital (m-) sequences were introduced in typical tables using hypothesis test using Chi-square test.

The binary and digital poker test results are obtained by program using version 10.0 of MATLAB Language with no time mentioned to obtain the randomness results.

II. Hypothesis Test for Randomness Using Chi-Square Test

Suppose we have an experiment with n ≥ 2 possible outcomes, with unknown probabilities p_1, p_2, ..., p_n, now we attempt to decide which of the two hypotheses H_0 or H_1 is true, where H_0: p_i = p_0 for all i=1,2,...,n and H_1: p_i ≠ p_0 for some i=1,2,...,n, where p_0, p_0, ..., p_0 are known.

DOI: 10.9790/5728-12151723 www.iosrjournals.org
We consider \( n \) independent repetitions of the experiment with the random variables \( N_i \) denoting the number of times the \( i \)th outcome occurs, for \( i=1,2,\ldots,n \) where \( \sum _{i=1}^{n} N_i = L \). We use the test statistic \( W \). In particular, a small value of \( W \), where \( W \) is the observed value of \( \chi^2 \), is evidence in favor of \( H_0 \) (provided that the data are really generated from a random sample).

In order to test the hypothesis, we applied the following steps [8]:
- State the null and alternative hypotheses.
- Select the distribution to use.
- Determine the rejection and non-rejection regions.
- Calculate the value of the test statistic.
- Make a decision.

### III. Binary Poker Test

Let \( d \) be a positive integer such that \( d \geq 3 \), and let \( k = d \). Divide the sequences into \( k \) non-overlapping parts each of length \( d \), and let \( n_i \) be the observed number of occurrences of the \( i \)th type of sequence of length \( d \), \( 0 \leq i \leq d \). The poker test determines whether the sequences of length \( d \) each appear approximately the same number of times in \( S \), as would be expected for a random sequence. The expected value of the string which consists of \( i \) 1’s with no consideration to arrangement of (1’s) is [9]:

\[
E_i = C_i^d \cdot \frac{1}{2^n} \cdot \frac{L}{d}
\]

The statistic used is:

\[
\tilde{\chi}^2 = \sum \frac{(n_i - C_i^d \cdot \frac{1}{2^n} \cdot \frac{L}{d})^2}{C_i^d \cdot \frac{1}{2^n} \cdot \frac{L}{d}}
\]

which approximately follows a \( \chi^2 \) distribution with \( v = d \) degrees of freedom. Note that the poker test is a generalization of the frequency test: setting \( d = 1 \) in the poker test yields the frequency test.

**Example (1):** Let's have the following sequence with length \( L = 30 \), \( m = 2 \). Let \( S = 100 \ 111 \ 101 \ 110 \ 011 \ 001 \ 101 \ 001 \ 010 \ 000 \). For \( d = 3 \), \( C_i^d = 1,3,3,1 \), then \( E_i = 1.25,3.75,3.75,1.25 \) respectively. From the sequence we have the following frequencies using relation (3):

<table>
<thead>
<tr>
<th>( n_i )</th>
<th>Samples</th>
<th>Freq.</th>
</tr>
</thead>
<tbody>
<tr>
<td>( n_0 )</td>
<td>000</td>
<td>1</td>
</tr>
<tr>
<td>( n_1 )</td>
<td>001,010,001</td>
<td>4</td>
</tr>
<tr>
<td>( n_2 )</td>
<td>011,101,110</td>
<td>4</td>
</tr>
<tr>
<td>( n_3 )</td>
<td>111</td>
<td>1</td>
</tr>
</tbody>
</table>

\[
\tilde{\chi}^2 = \frac{(1 - 1.25)^2}{1.25} + \frac{(4 - 3.75)^2}{3.75} + \frac{(4 - 3.75)^2}{3.75} + \frac{(1 - 1.25)^2}{1.25} = 0.1333, \text{ note } \tilde{\chi}^2 = 0.1333 < \chi^2_0 = 7.81, \text{ this mean we accept } H_0 \text{ and reject } H_a, \text{ this sequence pass the binary poker test.}

**Example (2):** We test two sequences, \( S_1 \) which is generated from random generator (random function of the computer) while the sequence \( S_2 \) generated from non-random generator (two shift registers with AND boolean function) using binary or classical poker test (CPT) for \( m = 2 \) and \( d = 3,4,5 \). Table (1) includes the information of the length of the two sequences with the expected \( (E) \) and the frequencies \( (n_i) \) values. While table (2) shows the

DOI: 10.9790/5728-12151723
\( \hat{X}^2 \) values and the accepted or rejection decision for the two sequence \( S_1 \) and \( S_2 \), the information of applying the relation (3) obtained from table (1). \( \hat{X}^2 = 7.815, 9.488 \) and 11.071 for \( d=3,4 \) and 5 respectively.

**Table (1):** Information of the two sequences \( S_1 \) and \( S_2 \).

<table>
<thead>
<tr>
<th>( d )</th>
<th>( L=1000 )</th>
<th>( L=5000 )</th>
<th>( L=10000 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( n )</td>
<td>( E_1 )</td>
<td>( S_1 )</td>
<td>( S_2 )</td>
</tr>
<tr>
<td>3</td>
<td>41.67</td>
<td>125</td>
<td>125</td>
</tr>
<tr>
<td>4</td>
<td>15.63</td>
<td>62.5</td>
<td>93.75</td>
</tr>
<tr>
<td>5</td>
<td>6.25</td>
<td>31.25</td>
<td>62.5</td>
</tr>
</tbody>
</table>

**Table (2):** classical Poker test (CPT) results for two sequences.

<table>
<thead>
<tr>
<th>Seq.</th>
<th>CPT Values</th>
<th>( L )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \hat{X}^2 )</td>
<td>( \hat{X}^2 )</td>
<td>( \hat{X}^2 )</td>
</tr>
<tr>
<td>Decision</td>
<td>Accept</td>
<td>Accept</td>
</tr>
<tr>
<td>Decision</td>
<td>Reject</td>
<td>Reject</td>
</tr>
</tbody>
</table>

**Remark (2):** From example (1), if we re-calculate the frequency and expected the values for the samples of binary sequence:

<table>
<thead>
<tr>
<th>( n_{ijk} )</th>
<th>( n_{000} )</th>
<th>( n_{001} )</th>
<th>( n_{010} )</th>
<th>( n_{011} )</th>
<th>( n_{100} )</th>
<th>( n_{101} )</th>
<th>( n_{110} )</th>
<th>( n_{111} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Samples</td>
<td>000</td>
<td>001</td>
<td>010</td>
<td>011</td>
<td>100</td>
<td>101</td>
<td>110</td>
<td>111</td>
</tr>
<tr>
<td>Freq.</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

These samples have the same expected value s.t. \( E_i = (1/2)^i \cdot (L/d) = (1/2)^i \cdot (30/3) = 1.25 \). So relation (3) can be extended to following relation for \( d=3 \):

\[
\hat{X}^2 = \sum_{i=0}^{3} \sum_{j=0}^{i} \sum_{k=0}^{j} \frac{(n_{ijk} - E)^2}{E} \quad \text{(4)}
\]

\[
\hat{X}^2 = \left(1 - 1.25\right)^2 + \left(2 - 1.25\right)^2 + \cdots + \left(1 - 1.25\right)^2 = 1.25
\]

Notes the closed results for example (1) in relations (3) and (4).

In the next proposition we will prove that if the binary sequence \( S \) passes the poker test, \( \hat{X}^2 \leq \hat{X}^2 \), in relation (3), then it will satisfies poker test for relation (4) and vice versa.

**Proposition (1):** The sequence \( S \) passes binary poker test for relation (3) if and only if it passes binary poker test for relation (4).  

**Proof:**  
1 \( \rightarrow \) 2: this mean \( n_i \approx E_i = C_i p \), where \( p = (1/2d) (L/d) \). For simplicity let’s choose \( d=2 \), then:

\( n_0 = n_{00}, n_1 = n_{01} + n_{10} \) and \( C_0 = C_{00} = 1, C_1 = C_{01} + C_{10} = 1 + 1 = 2, C_2 = C_{11} = 1 \).
Applied Poker Test for General Digital Sequences

\[ \hat{X}^2 = \sum_{i=0}^{2} \left( \frac{(n_i - C_i \cdot p)^2}{C_i \cdot p} \right) = \frac{(n_0 - C_0 \cdot p)^2}{C_0 \cdot p} + \frac{(n_1 - C_1 \cdot p)^2}{C_1 \cdot p} + \frac{(n_2 - C_2 \cdot p)^2}{C_2 \cdot p} \]

\[ \frac{(n_0 - C_0 \cdot p)^2}{C_0 \cdot p} = \frac{(n_{00} - C_{00} \cdot p)^2}{C_{00} \cdot p} = \frac{(n_{00} - p)^2}{p} \]

...(a)

and

\[ \frac{(n_2 - C_2 \cdot p)^2}{C_2 \cdot p} = \frac{(n_{11} - C_{11} \cdot p)^2}{C_{11} \cdot p} = \frac{(n_{11} - p)^2}{p} \]

...(b)

Now

\[ \frac{(n_1 - C_1 \cdot p)^2}{C_1 \cdot p} = \frac{(n_{01} + n_{10} - (C_{01} + C_{10}) \cdot p)^2}{(C_{01} + C_{10}) \cdot p} = \frac{(n_{01} + n_{10} - 2p)^2}{2p} \]

\[ = \frac{(n_{01} - p)^2 + (n_{10} - p)^2}{2p} + \left[ \frac{1}{p} \cdot (n_{01} - p) \cdot (n_{10} - p) \right] \]

...(c)

Since \( n_i \approx C_i p \), then

\[ |n_{01} + n_{10}| \approx |(C_{01} + C_{10}) \cdot p| = |p + p| \]

\[ |n_{01} - p| \approx |n_{10} - p| \]

\[ (n_{01} - p) \cdot (n_{10} - p) \approx \frac{1}{2} \left[ (n_{01} - p)^2 + (n_{10} - p)^2 \right] \]

...(d)

Substitute (d) in (c), obtain:

\[ \frac{(n_1 - C_1 \cdot p)^2}{C_1 \cdot p} \approx \frac{(n_{01} - p)^2 + (n_{10} - p)^2}{p} \]

...(e)

From (a), (b) and (e) we obtain:

\[ \hat{X}^2 = \sum_{i=0}^{d} \left( \frac{(n_i - C_i \cdot z^i \cdot \frac{1}{z^i} \cdot \frac{1}{d})^2}{C_i \cdot \frac{1}{z^i} \cdot \frac{1}{d}} \right) \approx \sum_{i=0}^{1} \sum_{j=0}^{1} \sum_{k=0}^{1} \frac{(n_{jk} - E)^2}{E} \]

In the same way we can prove the other side.

Proposition (1) can be proved for any \( d \), and now its no matter if we applied relation (3) or (4) to decide whether that \( S \) will passes binary poker test.

IV. Generalizing the Binary \( d \)-Tuple Digital Poker Test

In this section we will generalize the binary \( d \)-tuple sequences to calculate the digital poker test for digital \( (m) \)-sequences.

Let \( S \) be the digital \( m \)-sequence generated from digital generator with length \( L \), where \( s_i \in S \), and \( 0 \leq s_i \leq m - 1 \), \( i=0,1,2,...,L-1 \).

Now we will propose a new poker test, we called it Grouped poker test (GPT), s.t. we will divide the samples of \( d \)-tuple subsequence to \( m \) disjoint sets of samples. Let \( A=\{0,1,...,m^d-1\} \) be the set of all different integer sum of the all probable \( d \)-tuple samples of the tested digital sequence \( S \). We partitioned the set \( A \) into equal order subsets \( A_0=\{x_0=0,1,...,x_{1}\} \), \( A_k=\{x_1,x_1+1,...,x_{1-}\} \), ..., \( A_m=\{x_{m-1},x_{m-1}+1,...,x_{m}=m^d-1\} \), s.t. \( x_{i}=i(m^{d-1}) \), and \( i=0,1,...,m-1 \). Notice that:

DOI: 10.9790/5728-12151723  www.iosrjournals.org 20 | Page
Applied Poker Test for General Digital Sequences

- \( A = \bigcup_{j=0}^{m-1} A_j \).
- \( A_k \cap A_j \), where \( k \neq j \), \( 0 \leq k, j \leq m-1 \) (any two different subsets of \( A \) are disjoint).
- \( |A| = m^d \) and \( |A_i| = h = m^d - 1 \), \( 0 \leq i \leq m-1 \).

**Example (3):** For \( m=3 \) and \( d=3 \), we have the following samples:
- \( (0,0,0),(0,0,1),(0,0,2),(0,1,0),(0,1,1),(0,1,2),(0,2,0),(0,2,1),(0,2,2), \)
- \( (1,0,0),(1,0,1),(1,0,2),(1,1,0),(1,1,1),(1,1,2),(1,2,0),(1,2,1),(1,2,2), \)
- \( (2,0,0),(2,0,1),(2,0,2),(2,1,0),(2,1,1),(2,1,2),(2,2,0),(2,2,1),(2,2,2), \)

This means we have \( m^3 = 27 \) samples. We have \( A_0 = \{0,1,\ldots,8\} \), \( A_1 = \{9,10,\ldots,17\} \) and \( A_2 = \{18,19,\ldots,26\} \). \( |A| = 27 \), \( |A_i| = 3^2 = 9 \), \( m = 0,9,18, i=0,1,2 \).

Now let \( N_i, i=0,1,\ldots,m-1 \), be the total observed frequency of all elements of the set \( A_i \) occurred in digital sequence \( S \). The probability \( P \) for each digit \( 0,\ldots,2^d-1 \) in the set \( A \) is:

\[
P(s) = 1/2^n, \quad k=0,1,\ldots,m^d-1.
\]

Since the order of the set \( A_i \), \( |A_i| = h = m^d - 1 \), then the probability for each digit in set \( A_i \) is:

\[
P(s) = h(1/2^n) = 1/m, t=i+1,\ldots,ih.
\]

Then the expected value \( E \) is:

\[
E = (1/m). (L/d) = L(m.d)
\]

which is equal for all digits \( s_i \).

When applying chi-square test, \( \chi^2 \) will compared with table value \( \chi^2 \) with freedom degree \( \nu = m \).

It's obvious that if the frequency \( N_i \) of each distinct digit \( s_i \) is approximate to other frequencies, then the digital sequence is satisfies the poker postulate, so must be:

\[ N_0 \approx N_1 \approx \ldots \approx N_{m-1} \]

Then the GPT will be:

\[
\chi^2 = \sum_{i=0}^{m-1} \left( \frac{N_i - \frac{1}{m.d}}{\frac{1}{m.d}} \right)^2
\]

**Example (4):** Let's call example (1), \( L=30, m=2, d=3 \). \( x_i = 0, 4, A_0 = \{0,1,2,3\} \) and \( A_1 = \{4,5,6,7\} \), \( E = 0.5 \times (30/3) = 5 \),

<table>
<thead>
<tr>
<th>( N_i )</th>
<th>Samples</th>
<th>Freq.</th>
</tr>
</thead>
<tbody>
<tr>
<td>( N_0 )</td>
<td>000,001,010,100</td>
<td>5</td>
</tr>
<tr>
<td>( N_1 )</td>
<td>100,101,110,111</td>
<td>5</td>
</tr>
</tbody>
</table>

\[
\chi^2 = \sum_{i=0}^{m-1} \left( \frac{N_i - \frac{1}{m.d}}{\frac{1}{m.d}} \right)^2 = \left( \frac{5 - 5}{5} \right)^2 + \left( \frac{5 - 5}{5} \right)^2 = 0
\]

\( \chi^2 = 0 < 7.815 \), then 5 passes GPT.

**Example (5):** Let's call example (2), using GPT for \( m=2 \) and \( d=3,4,5 \). Table (3) includes the new information of the two sequences \( S_1 \) and \( S_2 \) with the new expected \( E \) and new frequencies \( N_i \) values. While table (4) shows the \( \chi^2 \) values and the accepted or rejection decision for the sequence \( S_1 \) and \( S_2 \), the information of applying the relation (8) obtained from table (3).

\( \chi^2 = 5.991 \) for \( \nu = m = 2 \).

DOI: 10.9790/5728-12151723 www.iosrjournals.org 21 | Page
Generalize the Randomness Tests to Test the Digital Sequences Produced

A Computer Package for Measuring the Strength of Encryption

Probability, Markov Chains, Queues, and Simulation: The Mathematical Ba

Example (6): Now let’s choose \( m = 3, 4 \) and 5, and \( d = 3, 4, 5 \) for the two sequences \( S_1 \) and \( S_2 \) with length \( L = 5000 \) digits. Table (5) illustrates the GPT results using chi-square test for the two chosen sequences. \( \chi^2 \) holds as long as possible and choose many examples of sequences to be tested by poker test to obtain accurate decision.

In order to obtain accurate randomness decision for any digital sequence, we have to choose different values for \( d \).

From Ali et al. [3] Ibrahim [7] with this work we will obtain five digital randomness test for digital sequences so we suggest a digital statistical randomness package using five digital tests; frequency, run, autocorrelation, serial and poker tests.

V. Conclusions

1. As known, the proof of randomness for any sequence is probabilistic, so it’s preferable to choose \( L \) as long as possible and choose many examples of sequences to be tested by poker test to obtain accurate decision.

References


DOI: 10.9790/5728-12151723 www.iorsjournals.org 22 | Page