

A Statistical Measurement of Randomness Based on Pattern Vectors

Ray-Ming Chen

Baise University

21, Zhongshan No.2 Road, Guangxi Province, 533000

China

Received: March 18, 2020. Revised: April 20, 2020. 2nd Revised: May 25, 2020.

Accepted: June 23, 2020. Published: July 20, 2020.

Abstract- Randomness of data or signals has been applied and studied in various theoretical and industrial fields. There are many ways to define and measure randomness. The most popular one probably is the statistical testing for randomness. Among the approaches adopted, Runs Test is a highly used technique in testing the randomness. In this article, we demonstrate the inefficient aspects of Runs Test and put forward a new approach, or pattern-vector-based statistic, based on pattern vectors that could effectively enhance the precision of testing randomness. A random binary sequence is supposedly to have less or no patterns. Based on this, we put forward our randomness-testing statistic. We also run an experiment to demonstrate how to apply this statistic and compare the efficiency or failure rate with Runs Test in dealing with a set of randomly generated input sequences. Moreover, we devise a statistically-justifiable measure of randomness for any given binary sequence. In the end, we demonstrate a way to combine this new device with Kalman filters to enhance the data assimilation.

Keywords- randomness, pattern vectors, inner products, Kalman Filter

I. INTRODUCTION

Randomness of a sequence generated from an experiment is a vital and open question. Its openness lies in the need of full comprehension and definition of "randomness". Nonetheless, the practical usages arising from various applications [2] - for example, testing the fairness of a lottery machine, a slot machine, a dice, a peck of poker, etc. - spurs the development of finding suitable measures for randomness. There are many theoretical measurements and statistical measurements of randomness [3]. In physics, the usual one is the concept of entropy[4, 5]. For the statistical ones, a convenient and widely-used approach is the Runs Test, which is largely regarded as the standard statistical test for randomness of sequences in

the textbooks [1]. In this article, we demonstrate that Runs Test indeed is quite inaccurate and too simplistic in randomness testing. We put forward a new approach based on pattern recognitions (vectors) to capture the randomness of a sequence. On the other hand, inner products [7, 8] or semi-inner products [6] are usually used to delve into the relationship between two set of data. In this article, we choose some patterns to induce pattern vectors to capture the similarities between the input vector (sequence) and the pattern vectors. Then we devise a statistic to measure the randomness via the concept of variance between the degrees of similarities.

II. RUNS TEST

The statistical inference regarding the randomness of a sequence is:

H_0 : the experimental binary sequence generated is random.

H_1 : the experimental binary sequence generated is not random.

Here we review and rephrase some terminologies and formulas for Runs Test of randomness [1, 9]. For consistent computation in this article, we use n_{-1} (rather than n_0) and n_1 to denote the usual meaning of the number of occurrence of 0 and 1 (or positive/negative signs) in a binary sequence, respectively. We use *Runs* to denote Runs Test statistic and x to denote the number of runs in a sequence. Then the distribution of Runs Test statistic *Runs* goes as follows: $P(Runs = x|H_0) =$

$$\begin{cases} 2 \cdot \frac{\binom{\frac{n_{-1}-1}{2}}{\frac{x}{2}-1} \cdot \binom{\frac{n_1-1}{2}}{\frac{x}{2}-1}}{\binom{n_{-1}+n_1}{n_{-1}}}, & \text{if } x \text{ is even;} \\ \frac{\binom{\frac{n_{-1}-1}{x+1}}{\frac{x-1}{2}} \cdot \binom{\frac{n_1-1}{x-1}}{\frac{x-1}{2}} + \binom{\frac{n_{-1}-1}{x+1}}{\frac{x-1}{2}} \cdot \binom{\frac{n_1-1}{x-1}}{\frac{x-1}{2}}}{\binom{n_{-1}+n_1}{n_{-1}}}, & \text{if } x \text{ is odd.} \end{cases}$$

In the latter chapter, we will compare the efficiency of Runs Test via this formula with our pattern-vector-based statistic. Before that, let us build a sound theory

regarding non-numerical data in order to accommodate our construction of models.

III. FORMALIZATION OF SET-BASED INNER PRODUCT

Since binary sequences normally doesn't have numerical meaning, for examples, a sequence of head-and-tail tosses of a coin, and connect more to nominal values, we formalize a system that could accommodate nominal values via set-based mechanisms. We embed a copy of numerical inner product with integer coefficients into Cantorian set theory. By doing so, we could analyse non-numerical data by numerical tools, like Euclidean inner product.

Therefore, in this section, we give an approach to embed the concept of inner product into Cantorian set theory. Though it is also possible to formalize them in multi-set theory, it would be cumbersome in dealing the concept of negative multiplicities. Hence, we resort to the original Cantorian set theory to build up a formalized version of inner product. The reason for us to embed numerical inner product into sets is to link the intrinsic properties shared by both numerical data and non-numerical data, which is usually treated qualitatively, rather than quantitatively. Let $U = \{a_1, a_2, \dots, a_n\}$ denote a set of target set (universe), where $n = |U|$. U is not restricted to any numerical set. It could be any data type, for example, characters, letters, numbers, sentences, etc. In order to accommodate the multiplicity of a set A , we define a function $m_A : U \rightarrow \mathbb{Z}$ and embed each multi-set A into Cantorian set via $A \equiv \bigcup_{p=1}^n \bigcup_{i=0}^{m_A(a_p)} \{(a_p, i)\}$ or

$$A \equiv \bigcup_{i=0}^{m_A(a_1)} \{(a_1, i)\} \bigcup_{i=0}^{m_A(a_2)} \{(a_2, i)\} \dots \bigcup_{i=0}^{m_A(a_n)} \{(a_n, i)\}.$$

Let MS_U denote the set of all the multi-sets defined over U . Let $A, B, C \in MS_U$ be arbitrary. In the following, we equip all the operations for the structure $(MS_U, \otimes, \times, \oplus, -, \ominus, \emptyset, \mathbb{Z} \otimes, \langle, \rangle)$. Observe that the scales for sets in MS_U are integers, which is sufficient in constructing our model.

The empty set in MS_U is identified by

$$\emptyset = \bigcup_{j=1}^n \{(a_j, 0)\}.$$

Definition III.1.

$$A \oplus B := \bigcup_{p=1}^n \bigcup_{i=0}^{m_A(a_p)+m_B(a_p)} \{(a_p, i)\}.$$

The product of two sets A and B in MS_U is identified as follows:

Definition III.2.

$$A \times B := \bigcup_{p=1}^n \bigcup_{q=1}^n \bigcup_{i=0}^{m_A(a_p)} \bigcup_{j=0}^{m_B(a_q)} \{((a_p, i), (a_q, j))\}.$$

We extend the usual multiplicity to negative integers for MS_U as follows.

Definition III.3. (Black Set) $-A := \bigcup_{p=1}^n \bigcup_{i=0}^{-m_A(a_p)} \{(a_p, i)\}.$

The interpretation of this set could be plausibly regarded as a black hole in set, in which the elements are hidden.

Definition III.4. $A \ominus B := A \oplus (-B).$

Definition III.5. $k \otimes A := \bigcup_{p=1}^n \bigcup_{i=0}^{k \cdot m_A(a_p)} \{(a_p, i)\}$ for all $k \in \mathbb{N}.$

Claim 1. Suppose $\tilde{A} = n \otimes A$. Then $m_{\tilde{A}}(a_i) = k \cdot m_A(a_i)$ and $m_{A \oplus B}(a_i) = m_A(a_i) + m_B(b_i)$ for all $1 \leq i \leq m.$

Proof. □

In the following, we define the counterpart of set-based inner product.

Definition III.6. (set-based product) Define

$$\langle A, B \rangle := \sum_{p=1}^n | \{ ((a_p, i), (a_p, j)) : ((a_p, i), (a_p, j)) \in A \times B \} |.$$

This definition simulates the numerical product via set Cantorian direct product.

Claim 2.

$$\langle A, B \rangle = \sum_{i=1}^n m_A(a_i) \cdot m_B(a_i).$$

Proof. By observing Definition III.6, the number of the direct product is exactly the multiplication of the multiplicities of set A and B . □

Lemma III.1. \langle, \rangle is an integer inner product on MS_U .

Proof. For the positive definiteness and symmetry properties, both follow immediately from Claim 2. Here we show the linearity property. Let $u, v \in \mathbb{Z}$ be arbitrary.

Then $\langle u \otimes A \oplus v \otimes B, C \rangle = \sum_{i=1}^n m_D(a_i) \cdot m_C(a_i)$, where $D = u \otimes A \oplus v \otimes B$. Then by Claim 1, $m_D(a_i) = u \cdot m_A(a_i) + v \cdot m_B(a_i)$, we have

$$\begin{aligned} \langle u \otimes A \oplus v \otimes B, C \rangle &= \sum_{i=1}^n (u \cdot m_A(a_i) + v \cdot m_B(a_i)) \cdot m_C(a_i) \\ &= u \cdot \sum_{i=1}^n m_A(a_i) \cdot m_C(a_i) + v \cdot \sum_{i=1}^n m_B(a_i) \cdot m_C(a_i) \\ &= u \cdot \langle A, C \rangle + v \cdot \langle B, C \rangle. \end{aligned}$$

□

To be precise, we are talking about the isomorphic-structure sense of inner product, where numerical product \cdot is identified with scale product of set, or \otimes and numerical $+$ is identified with scale addition of set, or \oplus . Since we have constructed the theoretical parts of our model construction, we could safely use all the usual sense of numerical inner products in this article. These is no need to complicate our model by using all the languages constructed in this section.

IV. PATTERN-VECTOR-BASED MEASUREMENT OF RANDOMNESS

In this section, we introduce our method in testing randomness. The method will be named pattern-vector-based statistic.

A. Introduction: Pattern-Vector-Based Statistic

Let $\vec{a} \in \{-1, 1\}^n$ denote an arbitrary (input) vector with length $n \in \mathbb{N}$ which indicates the number of experiments. Let $\mathbb{P} = \{\vec{P}_1, \vec{P}_2, \dots, \vec{P}_m\} \subseteq \{-1, 1\}^n$ be a set of pattern vectors. In this article, unless otherwise specified, we interchangeably use patterns (sub-vectors) and pattern vectors (repeating sub-vectors till the length of experiment). For example $(-1, 1)$ is a pattern, and $(-1, 1, -1, 1, -1, 1, -1, 1, -1, 1, -1, 1, -1, 1)$ is a pattern vector, based on the pattern $(-1, 1)$ up to the length of experiment 17. Let P_{ij} denote the j th element in the pattern vector \vec{P}_i . Let P denote the matrix $[P_{ij}]$, where $1 \leq i \leq m$ and $1 \leq j \leq n$. Let K^t denote the transpose of matrix K . Since we have justified inner product for applying in non-numerical data in Section III., we could construct a concrete model in this section via inner product. Let $\langle \cdot, \cdot \rangle_E$ denote the Euclidean dot inner product. Now we define the characterization of an input vector \vec{a} via the pattern vectors \mathbb{P} . We call these characterizations the coefficients of \vec{a} via pattern \mathbb{P} (or $CF_{\mathbb{P}}(\vec{a})$). Our method could be summarised in the following figure: The whole inference processes could be demonstrated in Figure 1. In the figure A is the ma-

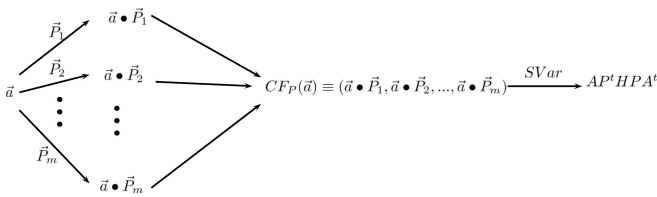


Fig. 1: pattern-vector-based Statistic

trix form of $veca$ and P is the matrix form of \mathbb{P} . $SVar$ and $AP^t HPA^t$ is the derived result and will be detailed in the following. Let us briefly take a look at this diagram. First of all, when an input vector (sequence) \vec{a} is received, its relation (coefficients) with other chosen pattern vectors $\mathbb{P} = \{\vec{P}_1, \vec{P}_2, \dots, \vec{P}_m\}$ is constructed and recorded in a coefficient vector $CF_{\mathbb{P}}(\vec{a})$ and then the variance of $CF_{\mathbb{P}}(\vec{a})$ is computed via a function $SVar$. The basic idea is if \vec{a} is a random sequence, then it tends to

have less or no preference over the choices of any particular pattern vectors. It then implies that the variance of coefficient vector or the variance of $CF_{\mathbb{P}}(\vec{a})$ tends to be smaller. Hence, a criterion to reject H_0 is when the variance is large. We have constructed our modelling for testing randomness. In the following, we detail all the terminologies and methods mentioned here.

B. Details and Properties Derivations

Definition IV..1. Define $CF_{\mathbb{P}} : \{-1, 1\} \rightarrow \mathbb{Z}^m$ by

$$CF_{\mathbb{P}}(\vec{a}) = (\langle \vec{a}, \vec{P}_1 \rangle_E, \langle \vec{a}, \vec{P}_2 \rangle_E, \dots, \langle \vec{a}, \vec{P}_m \rangle_E) \equiv (C_1, C_2, \dots, C_m),$$

$$\text{where } C_k := \langle \vec{a}, \vec{P}_k \rangle = \sum_{i=1}^n a_i \cdot P_{ki} \text{ for all } 1 \leq k \leq m.$$

Definition IV..2. $SVar(\vec{a}; \mathbb{P}) := \frac{\sum_{k=1}^m (C_k - \bar{C})^2}{m-1}$,

$$\text{where } \bar{C} = \frac{\sum_{k=1}^m C_k}{m}$$

Theorem IV..1. $SVar(\vec{a}; \mathbb{P}) = AP^t HPA^t$, where $H =$

$$\begin{bmatrix} \frac{1}{m} & \frac{-1}{m^2-m} & \dots & \frac{-1}{m^2-m} \\ \frac{-1}{m^2-m} & \frac{1}{m} & \dots & \frac{-1}{m^2-m} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{-1}{m^2-m} & \frac{-1}{m^2-m} & \dots & \frac{1}{m} \end{bmatrix} \text{ and } A = [a_1, a_2, \dots, a_n]$$

(i.e., \vec{a} in a matrix form).

Proof. $SVar(\vec{a}; \mathbb{P}) = \frac{\sum_{k=1}^m (C_k - \bar{C})^2}{m-1} = \frac{1}{m-1} \cdot [\sum_{k=1}^m C_k^2 - \frac{(\sum_{k=1}^m C_k)^2}{m}]$. Since

$$(\sum_{k=1}^m C_k)^2 = (AP^t \hat{I})(AP^t \hat{I})^t = AP^t (\hat{I} \hat{I}^t) PA^t$$

and

$$\sum_{k=1}^m C_k^2 = (AP^t)(AP^t)^t = AP^t IPA^t,$$

$$SVar(\vec{a}; \mathbb{P}) = \frac{1}{m-1} \cdot [AP^t IPA^t - \frac{AP^t (\hat{I} \hat{I}^t) PA^t}{m}] = AP^t [\frac{1}{m-1} \cdot I - \frac{\hat{I} \hat{I}^t}{(m-1) \cdot m}] PA^t, \text{ where } I \text{ is a } m \text{ by } m \text{ identity}$$

$$\text{matrix and } \hat{I} = \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix}. \text{ Let } H = \frac{1}{m-1} \cdot I - \frac{\hat{I} \hat{I}^t}{(m-1) \cdot m}. \text{ The}$$

result follows immediately. \square

In the following, we demonstrate how to compute the results in Figure 1

C. Patterns and Pattern Vectors

The patterns we choose are the periodic functions. We use h to denote the variables of length of periodic functions. For each given h , one has 2^h patterns. In Table 1, we give some examples for the patterns (or sub-vectors).

h	Patterns (sub-vectors)
1	(1),(-1)
2	(-1,-1),(-1,1),(1,-1),(1,1)
3	(-1,-1,-1),(-1,-1,1),(-1,1,-1),(-1,1,1),(1,-1,-1),(1,-1,1),(1,1,-1),(1,1,1)

Table 1: Patterns up to $h = 3$

If one chooses the number q as the maximum length for the range of h , then the total chosen patten vectors would be $2^1 + 2^2 + \dots + 2^q = 2^{q+1} - 2$. Suppose the number of experiments is $n = 25$, then correspondingly, we form the following pattern vectors (see Table 2) based on patterns chosen.

D. Computational Demonstration

Suppose the input vector (sequence) is

$$\vec{a} = (1, 1, 1, -1, 1, -1, -1, 1, -1, 1, -1, -1, -1, 1, -1, -1, 1, -1, 1, 1, 1, -1, -1)$$

Then $\vec{a} \bullet \vec{P}_1 = -1$ and by the same analogies for other $\vec{a} \bullet \vec{P}_j$, one has

$$CF_{\mathbb{P}}(\vec{a}) = (-1, 1, 1, 1, -1, -1, 1, -7, 5, -3, 3, -5, 7, -1).$$

By computing the variance of the vector $CF_{\mathbb{P}}(\vec{a})$, one gets $SVar(\vec{a}; \mathbb{P}) = 13.3846$.

V. EXPERIMENTAL RESULTS

Based on our theories and models, we run an experiment to yield an efficiency of our statistic. Then we compare our efficiency with the one of Runs Test. The result shows our approach could hugely enhance the prediction of randomness given the input binary sequences are really random.

A. Random Inputs and Pattern Vectors

We use n_{-1} and n_1 to denote the fixed appeared number of -1 and 1 , respectively. Let $N = n_{-1} + n_1$ be the total length of the experiment. In this experiment, we adopt $n_{-1} = 10, n_1 = 11$ and $N = 21$. We adopt R-language command “sample(c(-1,1),replace=TRUE,size=N)” to generate all the 10000 samples. All the results are stored in a matrix named *MRS*. However, only some of the randomly generated samples whose length consist of $n_{-1} = 10$ and $n_1 = 11$. We single out these which satisfy $n_{-1} = 10$ and $n_1 = 11$ and store these data in a sub-matrix *QRS* (qualified random samples). We choose $q = 4$ and have 30 patterns (or sub-vectors). Then we use these patterns to induce corresponding pattern vectors, and store these vectors in a 30 by 21 matrix *Pat*.

B. Experimental Space and Inferences

We use “expand.grid(rep(list(c(-1,1)),N))” to generate all the experimental samples and store them in

a matrix named *S*. However, not all the samples are valid as we have to restrict them to the samples with $n_{-1} = 10$ and $n_1 = 11$. After filtering out the unqualified samples, the remaining ones are stored in a matrix named *SS*. Then we calculate the range of our statistic which computes the variances of elements between *SS* and *Pat*. The range of our statistic is stored in a matrix named *RangePat*. Based on this, we calculate the critical point with significant level $\sigma = 0.01$. So far, we have constructed the distribution of our pattern-vector-based statistic.

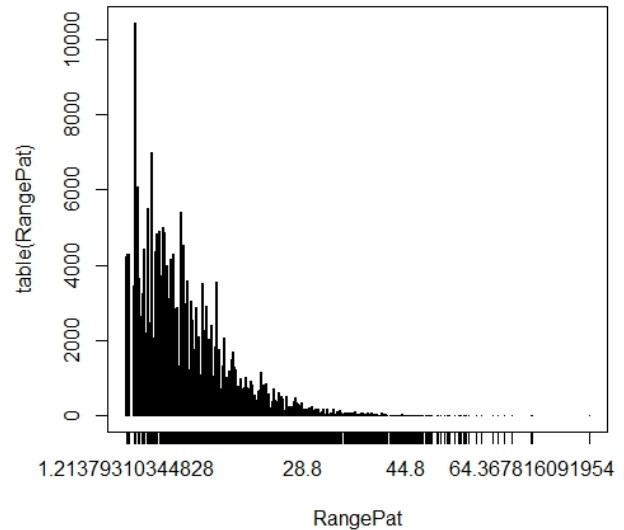


Fig. 2: Distribution of Pattern-Vector-Based Statistic

The critical point is then computed as 32.91954 as shown in Figure 2. Any input vector with higher value than this critical point should be regarded as an evidence to reject H_0 .

If the statement that H_0 : the experimental sequence is random is true, then by our statistic, there are 12 rejected cases out of $|QRS| = 1670$. The failure rate (or $\frac{12}{1670}$), which is defined to be the number of wrongly-rejected samples with respect to total valid samples is 0.007186. In comparison, the critical points, computed via formula in Section II. for Runs Test are 4 and 16, based the same significant level. However, the failure rate (or $\frac{56}{1670}$) of Runs Test is 0.033533, which is even higher than its significant level $\alpha = 0.01$ as shown in Figure 3. This means pattern-vector-based statistic is

Patterns	Pattern Vectors (repeated full vectors)
(1)	$\vec{P}_1 \equiv (1,1)$
(-1)	$\vec{P}_2 \equiv (-1,-1)$
(-1,-1)	$\vec{P}_3 \equiv (-1,-1)$
(-1,1)	$\vec{P}_4 \equiv (-1,1,-1,1,-1,1,-1,1,-1,1,-1,1,-1,1,-1,1,-1,1,-1,1,-1,1,-1,1,-1,1,-1,1,-1,1,-1)$
(1,-1)	$\vec{P}_5 \equiv (1,-1,1,-1,1,-1,1,-1,1,-1,1,-1,1,-1,1,-1,1,-1,1,-1,1,-1,1,-1,1,-1,1,-1,1,-1,1)$
(1,1)	$\vec{P}_6 \equiv (1,1)$
(-1,-1,-1)	$\vec{P}_7 \equiv (-1,-1)$
(-1,-1,1)	$\vec{P}_8 \equiv (-1,-1,1,-1,1,-1,1,-1,1,-1,1,-1,1,-1,1,-1,1,-1,1,-1,1,-1,1,-1,1,-1,1,-1,1,-1,1)$
(-1,1,-1)	$\vec{P}_9 \equiv (-1,1,-1,1,-1,1,-1,1,-1,1,-1,1,-1,1,-1,1,-1,1,-1,1,-1,1,-1,1,-1,1,-1,1,-1,1,-1)$
(-1,1,1)	$\vec{P}_{10} \equiv (-1,1,1,-1,1,1,-1,1,1,-1,1,1,-1,1,1,-1,1,1,-1,1,1,-1,1,1,-1,1,1,-1,1,1,-1)$
(1,-1,-1)	$\vec{P}_{11} \equiv (1,-1,-1,1,-1,1,1,-1,1,1,-1,1,1,-1,1,1,-1,1,1,-1,1,1,-1,1,1,-1,1,1,-1,1,1)$
(1,-1,1)	$\vec{P}_{12} \equiv (1,-1,1,1,-1,1,1,1,-1,1,1,1,-1,1,1,1,-1,1,1,-1,1,1,1,-1,1,1,1,-1,1,1,1)$
(1,1,-1)	$\vec{P}_{13} \equiv (1,1,-1,1,1,1,-1,1,1,1,-1,1,1,1,-1,1,1,1,-1,1,1,1,1,-1,1,1,1,1,-1,1,1)$
(1,1,1)	$\vec{P}_{14} \equiv (1,1)$

Table 2: Pattern Vectors based on Patterns up to $n = 25$

4.66 times better than the Runs Test.

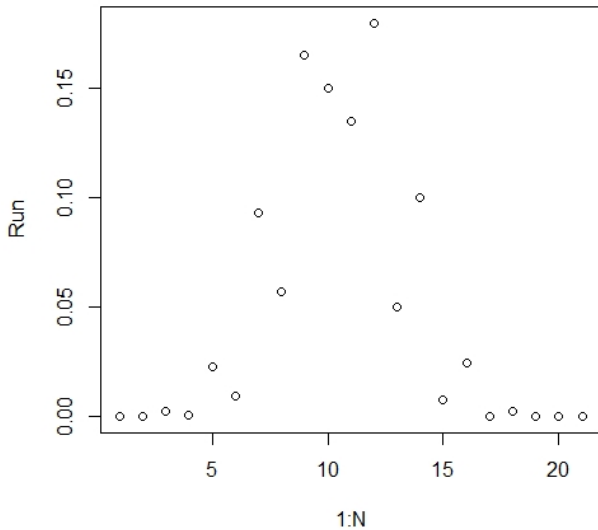


Fig. 3: Run Test Results

C. A New Measurement of Randomness

Since our method is statically justified, we could then define a measure that could be used to measure the randomness. We define the degree of randomness via a function $Deg : \{0, 1\}^n \rightarrow [0, 1]$ which is defined by

Definition V..1. $Deg(\vec{a}) := P(PVB \leq Var(\vec{a})|H_0)$, where PVB is our pattern-vector-based statistic.

VI. APPLICATION

In this section, we will demonstrate how to combine Kalman ([10, 11, 12]) Filters with our measurement of

randomness in dealing with multiple (simultaneous) sensors for the same target. Let x_t denote a (real) state vector with n state (random) variables. Let S_t denote the state transition matrix at (time) t . Let I_t denote the input dynamical matrix. Let c_t denote the control input vector. Let m_t denote the measurement vector. Let M_t denote the measurement matrix. Let n_t^x , and n_t^m denote state noise and measurement noise vector at t , respectively. Let Σ_t^x , and Σ_t^m denote process noise matrix and measurement noise matrix at t , respectively. Let \hat{x}_t^- and \hat{x}_t^+ denote the priori and posteriori estimate vector of X_t . A typical formulation of linear stochastic discrete time system goes as follows (here we omit all the typical assumptions):

- 1) $x_t = S_t x_{t-1} + I_t c_{t-1} + n_t^x$, where $n_t^x \sim \mathcal{N}(0, \Sigma_t^x)$;
- 2) $m_t = M_t x_t + n_t^m$, where $n_t^m \sim \mathcal{N}(0, \Sigma_t^m)$;
- 3) (state prediction) $\hat{x}_t^- = S_t \hat{x}_{t-1}^+ + I_t c_{t-1}$;
- 4) (error covariance prediction) $\hat{P}_t^- = S_t \hat{P}_{t-1}^+ S_t^T + \Sigma_t^x$, where $\hat{P}_t^- = E[(x_t - \hat{x}_t^-)(x_t - \hat{x}_t^-)^T]$.
- 5) (Kalman gain) $K_t = \hat{P}_t^- M_t^T (M_t \hat{P}_t^- M_t^T + \Sigma_t^m)^{-1}$.
- 6) (state update) $\hat{x}_t^+ = \hat{x}_t^- + K_t (m_t - M_t \hat{x}_t^-)$.
- 7) (error covariance update) $\hat{P}_t^+ = (I - K_t M_t) \hat{P}_t^-$, where I is the identity matrix.

Let us abuse the matrix notations and summarize the above results in the following:

- 1) (priori estimation) $\begin{bmatrix} \hat{x}_t^- \\ \hat{P}_t^- \end{bmatrix} = \begin{bmatrix} S_t & S_t(\cdot)S_t^T \\ I_t c_{t-1} \\ \Sigma_t^x \end{bmatrix} \begin{bmatrix} \hat{x}_{t-1}^+ \\ \hat{p}_{t-1}^+ \end{bmatrix} +$
- 2) (Kalman gain) $K_t = \hat{P}_t^- M_t^T (M_t \hat{P}_t^- M_t^T + \Sigma_t^m)^{-1}$.

$$3) \text{ (posteriori estimation) } \begin{bmatrix} \hat{x}_t^+ \\ \hat{p}_t^+ \end{bmatrix} = [I - K_t M_t \quad I - K_t M_t] \begin{bmatrix} \hat{x}_t^- \\ \hat{p}_t^- \end{bmatrix} + [K_t \quad K_t] \begin{bmatrix} m_t \\ 0 \end{bmatrix}$$

Now let us make the following assumptions regarding all the parameters:

- 1) $n = 2$;
- 2) $S_t = S = \begin{bmatrix} 1 & 0.5 \\ 0.6 & 1 \end{bmatrix}, I_t = \begin{bmatrix} 0 & 0.7 \\ 0.2 & 1 \end{bmatrix}, M_t = M = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \Sigma_t^x = \Sigma^x = \begin{bmatrix} 0.4 & 0.2 \\ 0.1 & 0.3 \end{bmatrix}, \Sigma_t^m = \Sigma^m = \begin{bmatrix} 0.1 & 0.5 \\ 0.1 & 0.2 \end{bmatrix}, c_{t-1} = c = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$;
- 3) $x_0 = \hat{x}_0^+ = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \hat{p}_0^+ = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$;

Then the above equations could be simplified as follows:

- 1) (state prediction) $\hat{x}_t^- = S\hat{x}_{t-1}^+$;
- 2) (error covariance prediction) $\hat{P}_t^- = S\hat{P}_{t-1}^+ S^T + \Sigma^x$.
- 3) (Kalman gain) $K_t = \hat{P}_t^- (\hat{P}_t^- + \Sigma^m)^{-1}$.
- 4) (state update) $\hat{x}_t^+ = \hat{x}_t^- + K_t(m_t - \hat{x}_t^-)$
- 5) (error covariance update) $\hat{P}_t^+ = (I - K_t)\hat{P}_t^-$.

In order to visualize the above relations, we abuse the matrix notations as follows:

- 1) (prediction) $\begin{bmatrix} \hat{x}_t^- \\ \hat{P}_t^- \end{bmatrix} = [S \quad S(\cdot)S^T] \begin{bmatrix} \hat{x}_{t-1}^+ \\ \hat{P}_{t-1}^+ \end{bmatrix} + \begin{bmatrix} 0 \\ \Sigma^x \end{bmatrix}$
- 2) (Kalman gain) $K_t = \hat{P}_t^- (\hat{P}_t^- + \Sigma^m)^{-1}$
- 3) (update) $\begin{bmatrix} \hat{x}_t^+ \\ \hat{P}_t^+ \end{bmatrix} = [I - K_t \quad I - K_t] \begin{bmatrix} \hat{x}_t^- \\ \hat{P}_t^- \end{bmatrix} + [K_t \quad K_t] \begin{bmatrix} m_t \\ 0 \end{bmatrix}$

Based on these relations, we start to implement them via R programs.

Since all the simulations will need a huge amount of programs, we specify the processes and leave the interested readers to fill the gaps. Suppose we sample 20 points (i.e., 20 sequential samples) from the bivariate normal distributions shown in Table 3. The second column shows the sampled process noise vectors $\{n_t^x | 1 \leq t \leq 20\}$. The third column shows the first batch of sampled measurement noise vectors $\{n_t^m | 1 \leq t \leq 20\}$ and the fourth column shows the second batch of sampled measurement noise vectors $\{n_t^m | 1 \leq t \leq 20\}$.

Then from $\{n_t^x | 1 \leq t \leq 20\}$, initial state vector in 3) and the recursive process in 1) (or $x_t = Sx_{t-1} + n_t^x$), we could simulate (or obtain) the actual state vectors x_t , i.e., $\{x_t | 1 \leq t \leq 20\}$. This, coupling with 2) (or $m_t = x_t + n_t^m$) and $\{n_t^m | 1 \leq t \leq 20\}$ would result in

$\{m_t | 1 \leq t \leq 20\}$, the (simulated) actual measurements. Let us use $\{m_t^{(1)} | 1 \leq t \leq 20\}$ and $\{m_t^{(2)} | 1 \leq t \leq 20\}$ to denote the actual measurement for batch one and batch two, respectively. If one looks at state update mechanism, he finds that m_t could be the weighted average of several finite batches of measurements (or sensors or observations). The weights could be decided via our measurement of randomness. We calculate $\{x_t - \hat{x}_t | 1 \leq t \leq 20\}$ and $\{m_t^k - \hat{m}_t^k | 1 \leq t \leq 20\}$, i.e., $\{m_t^k - \hat{x}_t^-\}$ in our case. One could expect that the higher the relation between the two set of vectors is, the higher weight should put upon the measurement (or sensors). Such relation is revealed via inner product $\{\langle x_t - \hat{x}_t, m_t^k - \hat{x}_t^- \rangle | 1 \leq t \leq 20\}$. Next, we convert this set into a sequence to 0 and 1, in which the value 0 is assigned whenever $\langle x_t - \hat{x}_t, m_t^k - \hat{x}_t^- \rangle \leq 0$ and 1, otherwise. We use δ_t^k to denote this value. Henceforth, we have an ordered set $\delta^k = (\delta_1^k, \delta_2^k, \dots, \delta_t^k, \dots, \delta_{20}^k)$. Based on the method we use, we could then measure the randomness of δ^k via $Deg(\delta^k)$ in Section V..1. We could then assimilate the measurements via v batches or sensors for

$$\text{the same target by } m_t = \sum_{k=1}^v \frac{Deg(\delta^k)}{\sum_{j=1}^v Deg(\delta^j)} \cdot m_t^k.$$

Now let us pose for while to analyse the Kalman filter. First of all, the white noise is assumed for n_t^x and n_t^m . It is hard to argue as n_t^x is accompanied with the dynamical equations and that is hard to detect, since we need to rely on measurement. Then the main factor that could justification the setting will heavily rely on n_t^m . However, this assumption could easily be examined by the real measurement. This is one problem that covariance $Cov(n_t^m)$ normally is hard to assume or calculate. In this section, we try to find a method that could preliminarily evaluate the effect of state estimation given the real measurement, rather than resorting to the simulated data, i.e., m_t will be measured directly.

VII. CONCLUSION

In this article, we have devised a new statistic named pattern-vector-based (PVB) statistic that, empirically, turns out to have higher predictive precision in testing the randomness of a sequence, in comparison with the usual Runs Test. The result might own to the fact that Runs Test doesn't really consider the interaction between different patterns and the studied input sequence. Our experimental results show that PVB could promote the accuracy from 4 to 5 times when compared with Runs Test. Based on this, we also design a statistically-justified measurement of randomness of binary sequence, which could then be coupled with Kalman Filters in dealing with data assimilation. Though we could enhance the precision in testing the randomness, the cost could not be ignored - the choices of patterns and pattern vectors would consume a lot of computational resources. An alternative is to pick up only those patterns that are much representative - for example, independence - rather than inclusively contain all the patterns based on length.

times	$\mathcal{N}(0, \Sigma^x)$	$\mathcal{N}(0, \Sigma^m) : 1$	$\mathcal{N}(0, \Sigma^m) : 2$
1	(-0.847 , -0.461)	(0.690 , -0.061)	(-0.251 , 0.495)
2	(0.380 , -0.049)	(0.976 , -0.176)	(0.645 , 0.760)
3	(-0.027 , 0.047)	(-0.253 , 0.527)	(-0.321 , -0.853)
4	(-1.304 , -0.140)	(-0.178 , -1.037)	(-1.304 , 0.255)
5	(-0.150 , -0.620)	(-0.451 , -0.134)	(0.343 , -0.248)
6	(0.811 , -1.051)	(0.760 , 0.259)	(0.634 , 0.321)
7	(-0.110 , -0.101)	(0.582 , 0.045)	(-0.866 , -0.750)
8	(0.053 , -0.881)	(-0.211 , -0.706)	(0.548 , -0.208)
9	(0.389 , -0.199)	(0.308 , -0.675)	(-0.020 , -0.254)
10	(-0.307 , 0.490)	(0.047 , 0.276)	(-1.294 , 0.541)
11	(-1.192 , 0.006)	(-0.665 , -0.242)	(0.223 , -0.381)
12	(-0.991 , -0.241)	(0.566 , 0.838)	(-0.900 , -0.062)
13	(-0.361 , -0.429)	(-0.784 , -0.750)	(0.764 , 0.149)
14	(0.014 , 0.223)	(-0.176 , 0.428)	(-0.947 , -1.205)
15	(-0.585 , -0.104)	(0.856 , -0.182)	(0.288 , 0.170)
16	(0.095 , 0.370)	(0.119 , -0.405)	(0.725 , 0.637)
17	(0.521 , 0.064)	(-0.055 , 0.035)	(0.772 , 1.019)
18	(0.020 , -0.495)	(1.174 , 0.395)	(0.354 , 0.485)
19	(-0.752 , -0.574)	(0.725 , 0.246)	(0.102 , 0.060)
20	(-0.235 , 0.456)	(0.433 , 0.510)	(-0.448 , -0.783)

Table 3: sampled data from bivariate normal distributions

ACKNOWLEDGMENT

This article is funded by the Humanities and Social Science Research Planning Fund Project under the Ministry of Education of China (Grant No. 20XJAGAT001).

REFERENCES

[1] Jean D. Gibbons, Subhabrata Chakraborti, Non-parametric Statistical Inference, 4th ed., Marcel Dekker Inc., New York, 2003.

[2] Jan Krhovjak, Vashek Matyas, and Jiri Zizkovsky, Generating Random and Pseudorandom Sequences in Mobile Devices, Security and Privacy in Mobile Information and Communication Systems, Springer, Germany, 2009.

[3] Raymond J. Hickey, A Note on the Measurement of Randomness, Journal of Applied Probability, Vol. 19, No. 1, 1982.

[4] C. E. Shannon , A mathematical theory of communication, The Bell System Technical Journal, Vol. 27, No. 3, 1948.

[5] Tarald O. Kvalseth ,On the Measurement of Randomness (Uncertainty) A More Informative Entropy, Entropy, MDPI, 2016.

[6] J. P. Antoine, K. Gustafson, Partial inner product spaces and semi-inner product spaces, Advances in Mathematics, Volume 41, Issue 3, 1981.

[7] A. Nagoor Gani, K. Kannan, A. R. Manikandan, Inner Product over Fuzzy Matrices, Journal of Mathematics, 2016.

[8] Loukas Grafakos. Modern Fourier Analysis. 2nd ed. Springer, 2009.

[9] Alhakim A. , Hooper W., A non-parametric test for several independent samples, Journal of Nonparametric Statistics, Vol. 20, No. 3, 2008.

[10] Kalman Rudolf, A New Approach to Linear Filtering and Prediction Problems, Transaction of the ASME—Journal of Basic Engineering, 1960.

[11] Eric Benhamou. Trend without hiccups: a kalman filter approach, IFTA Journal, 2017.

[12] Tony Lacey, Tutorial: The Kalman Filter, Access at <http://web.mit.edu/kirtley/kirtley/binlustuff/literature/control/Kalman>

Creative Commons Attribution License 4.0 (Attribution 4.0 International, CC BY 4.0)

This article is published under the terms of the Creative Commons Attribution License 4.0 https://creativecommons.org/licenses/by/4.0/deed.en_US