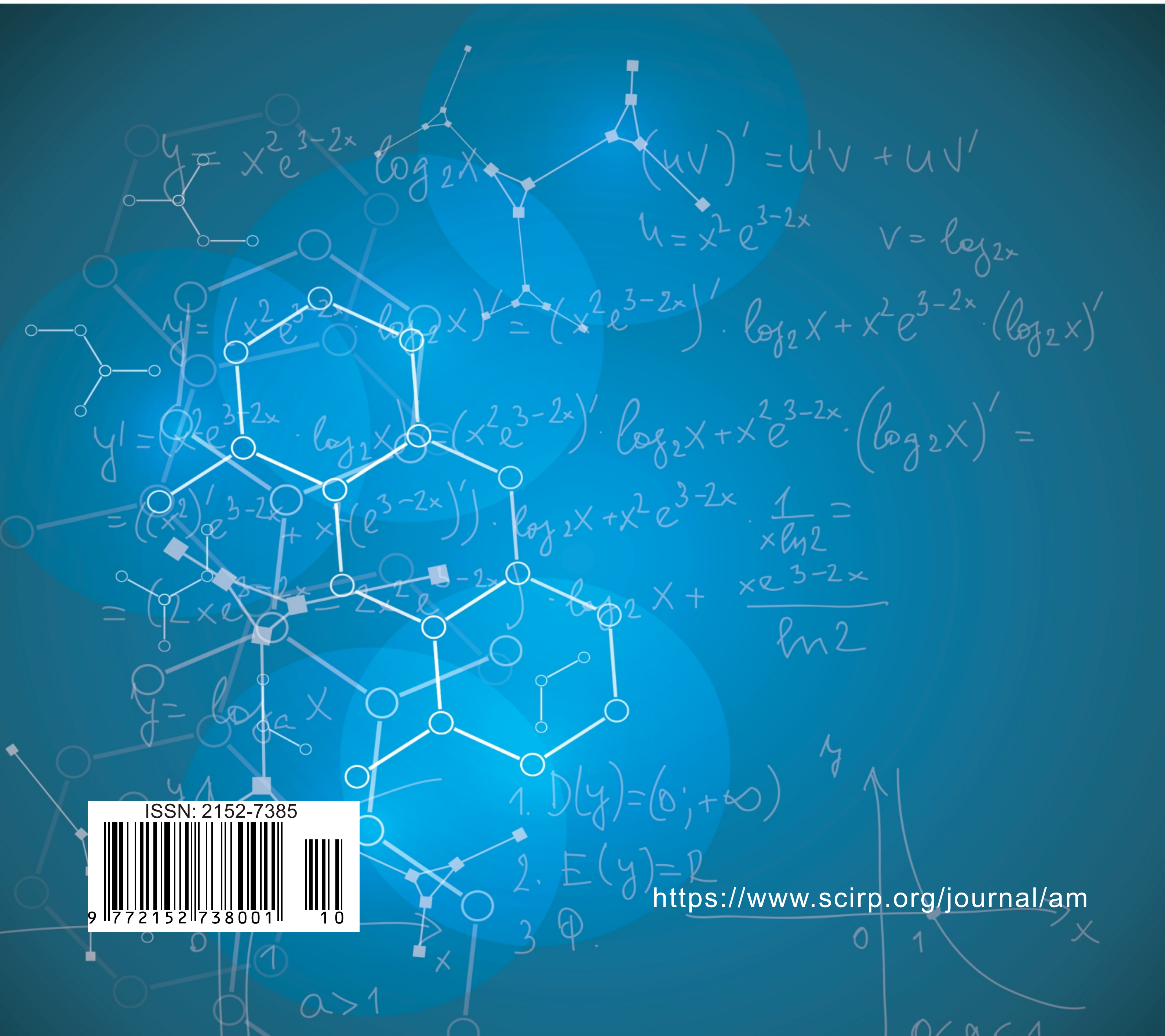
ISSN: 2152-7385 Volume 14, Number 10, October 2023

Scientific Research Publishing

Applied Mathematics



Journal Editorial Board

ISSN Print: 2152-7385 ISSN Online: 2152-7393

https://www.scirp.org/journal/am

Editorial Board

Applied Mathematics, 2023, 14, 673-717
https://www.scirp.org/journal/am
ISSN Online: 2152-7393
ISSN Print: 2152-7385

Table of Contents

Volume 14 Number 10 October 2023

Applied Mathematics (AM)
Journal Information

SUBSCRIPTIONS

The *Applied Mathematics* (Online at Scientific Research Publishing, https://www.scirp.org/) is published monthly by Scientific Research Publishing, Inc., USA.

Subscription rates:
Print: $89 per copy.
To subscribe, please contact Journals Subscriptions Department, E-mail: sub@scirp.org

SERVICES

Advertisements
Advertisement Sales Department, E-mail: service@scirp.org

Reprints (minimum quantity 100 copies)
Reprints Co-ordinator, Scientific Research Publishing, Inc., USA.
E-mail: sub@scirp.org

COPYRIGHT

Disclaimer of liability
Statements and opinions expressed in the articles and communications are those of the individual contributors and not the statements and opinion of Scientific Research Publishing, Inc. We assume no responsibility or liability for any damage or injury to persons or property arising out of the use of any materials, instructions, methods or ideas contained herein. We expressly disclaim any implied warranties of merchantability or fitness for a particular purpose. If expert assistance is required, the services of a competent professional person should be sought.

PRODUCTION INFORMATION

For manuscripts that have been accepted for publication, please contact:
E-mail: am@scirp.org

Applied Mathematics, 2023, 14, 673-683
https://www.scirp.org/journal/am
ISSN Online: 2152-7393
ISSN Print: 2152-7385

Type 2 Possibility Factor Rotation in No-Data Problem

Houju Hori

Nara Community, Tsubakikishi Shrine, Nara, Japan
Email: uemura0742@yahoo.co.jp

How to cite this paper: Hori, H. (2023) Type 2 Possibility Factor Rotation in No-Data Problem. *Applied Mathematics*, **14**, 673-683.
https://doi.org/10.4236/am.2023.1410039

Received: May 4, 2023
Accepted: October 5, 2023
Published: October 8, 2023

Abstract

Uemura [1] discovered a mapping formula that transforms and maps the state of nature into fuzzy events with a membership function that expresses the degree of attribution. In decision theory in no-data problems, sequential Bayesian inference is an example of this mapping formula, and Hori *et al.* [2] made the mapping formula multidimensional, introduced the concept of time, to Markov (decision) processes in fuzzy events under ergodic conditions, and derived stochastic differential equations in fuzzy events, although in reverse. In this paper, we focus on type 2 fuzzy. First, assuming that Type 2 Fuzzy Events are transformed and mapped onto the state of nature by a quadratic mapping formula that simultaneously considers longitudinal and transverse ambiguity, the joint stochastic differential equation representing these two ambiguities can be applied to possibility principal factor analysis if the weights of the equations are orthogonal. This indicates that the type 2 fuzzy is a two-dimensional possibility multivariate error model with longitudinal and transverse directions. Also, when the weights are oblique, it is a general possibility oblique factor analysis. Therefore, an example of type 2 fuzzy system theory is the possibility factor analysis. Furthermore, we show the initial and stopping condition on possibility factor rotation, on the base of possibility theory.

Keywords

Type 2 Fuzzy Events, Quadratic Mapping Formula, Stochastic Differential Equation in Fuzzy Event, Possibility Principal Factor Analysis, Possibility Oblique Factor Analysis, Initial and Stopping Condition

1. Introduction

Okuda *et al.* [3] constructed the decision rule under the fuzzy environment; however, this is an example of Bayes Decision Rule. Otherwise, Uemura [1] [4] [5] and Hori *et al.* [2] constructed another decision making on vague events. This decision-making is a special case for Bayes Decision Theory in No Data problem. In this paper, we obtain a mention of the system theory to possibility factor rotation according to the type 2 vague events.

Uemura [1] found a mapping formula that maps and transforms the state of nature in no information problem, (a no-data problem), in which no observable information can be observed in Bayesian statistics, to a fuzzy event by a membership function representing its degree of attribution. Note that the no-data problem can be attributed to Bayesian statistics, where the causality law between the state of nature and the observed information is uniform. Now, the fuzzy in this paper is sometimes called Vague to distinguish it from the Fuzziness of Zadeh [6]. Therefore, the extension of our study is named Vague Sets and Theory (Hori, Takemura, and Matsumoto [2]). Zadeh's fuzzy deals with vertical ambiguity, e.g., possibility interval type regression modeling, while our Vague deals with horizontal ambiguity, e.g., α-level cut of fuzzy sets. Also, Zadeh's modeling is conceptually very close to the interval-type modeling of subjective Bayesian theory, and the rotation based on our quadratic mapping formula is very relevant for factor analysis or independent component analysis (Hori [7]). First, Uemura [1] defined a mapping function from the state of nature for fuzzy events. Next, Hori *et al.* [2] showed that this definition is a formula. When the formula for the mapping function for fuzzy events is adapted to the theory of utility functions and developed into a decision-making method based on the utility function in fuzzy events, it can be applied to the case of the two-choice question (Uemura [1]). This is because a nondiscriminatory state in decision withholding arises, and Hori *et al.* [2] imposed an ergodic condition between the previous and next nondiscriminatory states and developed it into a Markov (decision) process in fuzzy events and derived a stochastic differential equation of it in fuzzy events. In this paper, Type 2 Fuzzy Events in which both horizontal and vertical ambiguity are considered at the same time, are discussed. Here, the quadratic mapping formula transforms a non-mapping function by relating it to two mapping functions and provides an orthogonal rotation to the function after the quadratic transformation. For Type 2 Fuzzy Events, orthogonal rotations from 0 to 180 degrees can be interpreted as cases where the longitudinal possibility error model and the transverse necessity error model are considered. The orthogonal rotation from 180 to 360 degrees can be interpreted as the case where the longitudinal necessity error model and the transverse possibility error model are taken into account. Here, what is measured by the possibility and necessity measures can be regarded as a kind of information content, and the possibility main factor rotation is provided to increase this information content. Note that Type 2 Fuzzy Events are attributed to a multidimensional possibility multivariable error model that takes into account the possibility and necessity of the longitudinal variable error model and the transverse variable error model. Furthermore, we show the initial and stopping conditions on possibility general factor rotation.

2. Mapping Formulas in Fuzzy Events

Uemura [1] defined the formula for mapping a function $f(x)$ by $g_1(x)$ as a formula (1). However, the definition is provided as a system theory, while as an example of application, a two-choice question is provided with respect to determinism. Later, it is shown that this definition is a formula (Hori, Takemura, and Matsumoto [2])

$$\mathrm{SUP}_{y=f(x)} g_1(x) = g_1\left(f^{-1}(y)\right) \tag{1}$$

The stochastic differential equation for a fuzzy event that represents the transition of the nondiscriminatory state regarding the decision withholding in the two-choice question based on the state of nature in sequential Bayesian inference is formulated as in Equation (2), and the Markov process in the fuzzy event is obtained as its solution as in Equation (3). The pole of the *S*-Markov process in the fuzzy event is the mapping formula for the fuzzy event in Equation (1). Here, ergodic conditions between each natural state are assumed. In addition, the monotonicity of the function *f* is a condition because it requires the existence of the first rank of the inverse function f^{-1} (Hori, Takemura, and Matsumoto [2]).

$$\frac{\mathrm{d}F}{\mathrm{d}t} = b\left(t, f_t^{-1}(y_t)\right) + \sigma\left(t, f_t^{-1}(y_t)\right) \cdot W_t \tag{2}$$

It is assumed that b is the mean term, σ the variance term, and W the error term in the equation of state for normal events.

$$F_t = L^{-1}\left(t, g_1\left(f^{-1}(y_t)\right)\right) \tag{3}$$

Here, L is the transition matrix of the Markov process of normal events.

Although this formula is subject to strict condition between Ergodic Conditions and monotonicity conditions of the function *f*, in natural states, it is able to be applied to the Go-Reserved Judgement Problem in the no-data problem, and is applicable to fuzzy stochastic differential equations for the transition of nondiscriminatory states concerning decision withholding in sequential Bayesian inference.

3. A Simultaneous Stochastic Differential Equation for Type 2 Fuzzy Events

Hori [7] [8] formulated the quadratic mapping formula as in Equation (4).

$$\mathrm{SUP}_{\substack{y=f(x) \\ Z=g_1\left(f^{-1}(y)\right)}} g_2(Z) = g_2\left(g_1^{-1}\left(f^{-1}(y)\right)\right) \tag{4}$$

Here, Equation (4) is a quadratic mapping formula that maps Equation (1), once again, by $g_2(x)$. The special property of the quadratic mapping formula is that it inverts 180 degrees when the mapped functions are equivalent, as in Equation (5). This indicates that this is a type of principal factor analysis. In statistical principal factor analysis, a 180-degree rotation requires two rotations of every 90 degrees. However, note that the quadratic mapping formula reverses 180 degrees in one rotation.

$$\text{if } g_1(\cdot)=g_2(\cdot) \text{ then } x=f^{-1}(y) \tag{5}$$

In this paper, the concept of time t into this quadratic mapping formula and derive a type 2 Markov process and its simultaneous fuzzy stochastic differential equation, albeit inverse.

First, if the transition matrix is L, the Markov process D_t is formulated as follows. (Takahashi [9])

$$D_t = L(t, x_t) \tag{6}$$

The type 1 fuzzy Markov process, which introduces the concept of fuzzy events, is derived in Equation (7), and the type 2 fuzzy Markov process is derived in Equation (8).

$$F_t = L^{-1}(t, g_1(x_t)) \tag{7}$$

$$\begin{aligned} F_{F_t} &= \underset{y_t=L^{-1}(t,g_1(f(x_t)))}{\text{SUP}} L^{-1}(t, g_2(x_t)) \\ &= L^{-1}\left(t, L^{-1}\left(t, g_2\left(g_1^{-1}\left(f^{-1}(y_t)\right)\right)\right)\right) \end{aligned} \tag{8}$$

Here, the Markov process in Equation (8) is derived from the following simultaneous fuzzy stochastic differential equations.

Equation (9) represents the change in the x-axis direction, and Equation (10) represents the change in the y-axis direction. x_t is a fuzzy variable in the horizontal direction and follows the fuzzy stochastic differential equation in Equation (9), and Z_t is a fuzzy variable in the vertical direction and follows the fuzzy stochastic differential equation in Equation (11).

$$\begin{cases} \dfrac{\mathrm{d}Z}{\mathrm{d}t} = m_1\left(t, g_{1t}\left(f_t^{-1}(Z_t)\right)\right) + \sigma_1\left(t, g_{1t}\left(f_t^{-1}(Z_t)\right)\right)\cdot W_{1t} & (9) \\ \dfrac{\mathrm{d}x}{\mathrm{d}t} = m_2\left(t, g_{2t}\left(f_t(x_t)\right)\right) + \sigma_2\left(t, g_{2t}\left(f_t(x_t)\right)\right)\cdot W_{2t} & (10) \end{cases}$$

Here, $Z_t = f_t(x_t)$, so Equation (10) is equivalent to Equation (11).

$$\frac{\mathrm{d}Z}{\mathrm{d}t} = m_2\left(t, g_{2t}(Z_t)\right) + \sigma_2\left(t, g_{2t}(Z_t)\right)\cdot W_{2t} \tag{11}$$

In the simultaneous fuzzy stochastic differential Equations (9) and (10), when the sum of the weights of each equation is 1, the type 2 Markov process of Equation (8) is derived. This means that the fuzzy event is a direct sum, which is closely related to the main factor analysis in Section 5.

4. Type 2 Possibility Principal Factor Rotation

Type 2 Fuzzy Events simultaneously encompass a two-dimensional necessity variable error model that considers longitudinal and transverse possibility errors. The 180-degree orthogonal rotation is the case of Equation (5), where possibility theory is applied to these possibility variable error models. Note that since both longitudinal and transverse fuzzy variables are considered, possibility theory is able to be applied. In this paper, particular attention to the measure of the size relationship of the fuzzy set is paid. Here, the possibility measure (POS) and the necessity measure (NES) are defined as followed (D. Dubois and H. Parade [10]). In addition, M and N are assumed to be Orthogonal Fuzzy Events with orthogonal degrees of attribution.

$$\mathrm{POS}(M \geq N) \triangleq \underset{U \geq V}{\mathrm{SUP}} \min\left(\mu_M(U), \mu_N(V)\right) \tag{12}$$

$$\mathrm{POS}(M > N) \triangleq \underset{U}{\mathrm{SUP}} \inf_{V \geq U} \min\left(\mu_M(U), \mu_N(V)\right) \tag{13}$$

$$\mathrm{NES}(M \geq N) \triangleq \inf_{U} \underset{V \leq U}{\mathrm{SUP}} \max\left(1-\mu_M(U), \mu_N(V)\right) \tag{14}$$

$$\mathrm{NES}(M > N) \triangleq 1-\underset{U \geq V}{\mathrm{SUP}} \min\left(\mu_M(U), \mu_N(V)\right) \tag{15}$$

The possibility principal factor rotation matrix for type 2 fuzzy is as follows:

$$\begin{bmatrix} x_{t+1} \\ Z_{t+1} \end{bmatrix} = \begin{bmatrix} \mathrm{POS}(M \geq N) & \mathrm{NES}(M > N) \\ \mathrm{NES}(M > N) & \mathrm{POS}(M \geq N) \end{bmatrix} \begin{bmatrix} x_t \\ Z_t \end{bmatrix} \tag{16}$$

In particular, note that in (16), when the possibility measure is 1, it is the identity matrix, and when the necessity measure is 1, it is the inversion matrix. Therefore, the possibility main factor rotation matrix in Equation (16) indicates that the sum of the weights of Equation (9) and (10) in the simultaneous fuzzy stochastic differential equation is 1 (Hori [7] [11]).

5. Initial and Stopping Condition in Type 2 Possibility Principal Factor Rotation

The initial condition and stopping condition for a normal Markov process are shown in [12]. Since we deal with horizontal ambiguity, we introduce the concept of quadratic possibility theory to the rotation according to a complex Markov process. The initial and stopping condition are shown in Equation (17), (18) and (19), (20), respectively. Where the rotation can start from (18) satisfying the initial condition (17). And the rotation can stop under (20) satisfying the stopping condition (19).

1) $\left(F_{10t}, F_{20t}\right) = \left(Z_{10t}, Z_{20t}\right)$ (17)

2) $\mathrm{POS}\left(\left(F_{1t}, F_{2t}\right) \geq \left(Z_{1t}, Z_{2t}\right) \mid x_0\right) \leq \mathrm{POS}\left(\left(DF_{1t}x_0, DF_{2t}x_0\right) \geq \left(DZ_{1t}x_0, DZ_{2t}x_0\right)\right)$

3) $\mathrm{NES}\left(\left(F_{1t}, F_{2t}\right) \geq \left(Z_{1t}, Z_{2t}\right) \mid x_0\right) \geq \mathrm{NES}\left(\left(DF_{1t}x_0, DF_{2t}x_0\right) \geq \left(DZ_{1t}x_0, DZ_{2t}x_0\right)\right)$

Where $\left(DF_{1t}x_0, DF_{2t}x_0\right) = \left(DX_{1t}, DZ_{2t}\right)$

(Starting State) $F_{10}(\cdot)_t = F_{20}(\cdot)_t$ (18)

1) $F_{ti0} = Z_{ti0}\ (i = 1,2)$

2) $\mathrm{POS}\left(F_{it} \geq Z_{it} \mid x_{i0}\right) \leq \mathrm{POS}\left(DF_{xi0} \geq DZ_{xi0}\right)(i = 1,2)$

3) $\mathrm{NES}\left(F_{it} \geq Z_{it} \mid x_{i0}\right) \geq \mathrm{NES}\left(DFx_{i0} \geq DZx_{i0}\right)(i = 1,2)$

Where $DFx_{i0} = DZx_{i0}\ (i = 1,2)$

Where F_{i0t} and $DF_{i0t}\ (i = 1,2)$ represents 2 complex events, and the quadratic possibility theory is applied. If the mapping function is equivalent, they invert 180-degree, and the initial condition and stopping condition is reversed. Note that the complex event become also one in a simulation like this.

1) $(F_{1t0}, F_{2t0}) = (Z_{1t0}, Z_{2t0})$ (19)

2) $\mathrm{POS}((F_{1t0}, F_{2t0}) \geq (Z_{1t}, Z_{2t})) \leq \mathrm{POS}((DF_{10}, DF_{20}) \geq (DZ_{10}, DZ_{20}))$

3) $\mathrm{NES}((F_{1t}, F_{2t}) \geq (Z_{1t}, Z_{2t})) \geq \mathrm{NES}((DF_{10}, DF_{20}) \geq (DZ_{10}, DZ_{20}))$

Where $(DF_{10}, DF_{20}) = (DZ_{10}, DZ_{20})$

(Stopping State) $F_{10}(\cdot)_t = F_{20}(\cdot)_t$ (20)

1) $F_{ti0} = Z_{ti0}\ (i = 1, 2)$

2) $\mathrm{POS}(F_{it} \geq Z_{it}) \leq \mathrm{POS}(DF_{i0} \geq DZ_{i0})\ (i = 1, 2)$

3) $\mathrm{NES}(F_{it} \geq Z_{it}) \geq \mathrm{NES}(DF_{i0} \geq DZ_{i0})\ (i = 1, 2)$

Where $DF_{i0} = DZ_{i0}$

6. Type 2 Possibility Oblique Factor Rotation

Assume that the fuzzy variables in the *x*-axis direction and the fuzzy variables in the *y*-axis direction are transformed into fuzzy events *N* and *M* on the natural state *S* by the membership functions $\mu_N(S)$ and $\mu_M(S)$ Note that the membership functions $\mu_N(S)$ and $\mu_M(S)$ are derived by the quadratic mapping formula as follows

$$\mu_N(S) = \Pi_N\left(\Pi_M^{-1}\left(f_1^{-1}(x)\right)\right) \tag{21}$$

$$\mu_M(S) = \Pi_M\left(\Pi_N^{-1}\left(f_2^{-1}(y)\right)\right) \tag{22}$$

Π_N and Π_M are the prior possibility distributions of the fuzzy variables *N* along the *x*-axis and *M* along the *y*-axis. The system functions are $x = f_1(S)$ and $y = f_2(S)$, respectively.

After the transformation, fuzzy event *N* and fuzzy event *M* that are not direct sums, as shown in the image in **Figure 1** (Uemura [4]), are discussed. Note that the sum of the membership functions representing the degree of attribution of fuzzy event *N* and fuzzy event *M* to the state of nature is less than or equal to 1. Therefore, the nondiscriminatory event F_e is automatically derived as follows. In decision theory, when the sum of the membership functions is less than or equal to 1, it is better to use the probability of the fuzzy event, and when the sum is greater than 1, it is better to use the probability measure of the fuzzy event (Uemura [13]).

Suppose that the possibility distribution $\Pi_{FK}(K = 1, \cdots, n)$ of two or more non-orthogonal Fuzzy Events is pre-set by Equation (21) and (22).

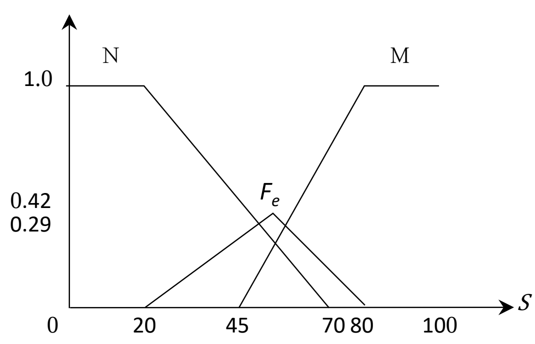


Figure 1. Indifferent event.

In this section, we consider the general case where $\sum_{k=1}^{n}\Pi_{Fk}(S)\le 1 \quad \forall_S \in S$. Here, we introduce the concept of Indifferent Event F_e in order to avoid the risk of decision-making arising from the lack of information in Fuzzy Events. The possibility distribution of this Indifferent Event can be automatically derived by the following equation.

$$\Pi_{F_e}(S)=1-\sum_k \Pi_{Fk}(S) \tag{23}$$

In this paper, we pick up the fuzzy variables in the *x*-axis direction and the fuzzy variables in the *y*-axis direction. Note that we consider only two fuzzy events such that $N = F_1$ and $M = F_2$.

The Indifferent Event F_e is divided into zones of the state of nature to make sense of it. In Zone $X=\{0\le s<20\}$, it is completely *N*, In Zone $a=\{20\le s<45\}$, it is a conditional Indifferent Event known to be a fuzzy event *N* In Zone $b=\{45\le s<70\}$, it is an Indifferent Event that is neither fuzzy event *N* nor fuzzy even *M*. However, the relationship between the magnitude of fuzzy event *N*. and fuzzy event *M*. Zone $c=\{70\le s<80\}$ is a conditional Indifferent Event that is known to be fuzzy event *M*. Zone $Y=\{80\le s\le 100\}$ is completely *M*. Here, each zone has different characteristics, so it is necessary to analyze each zone individually. However, decomposing and recomposing the system is very risky. In this decision problem, *N*, *M* and F_e are orthogonal sum events, so there is no need to decompose and recompose the system.

The fuzzy event *N* is derived from the stochastic differential Equation (9) along the *x*-axis, while the fuzzy event *M* is derived from the stochastic differential Equation (10) along the *y*-axis. With respect to the weights W_1 and W_2 of those simultaneous stochastic differential equations, if the two fuzzy events are in direct sum, that is, if the sum of the weights of each differential Equation (9) and (10) is orthogonal ($W_1 + W_2 = 1$), then it is a possible principal factor analysis (Hori [7] [11]). On the other hand, if the sum of the weights is less than 1 ($W_1 + W_2 < 1$), it is a possibility oblique factor analysis. In the following, for each zone, we focus on the indicator of the fuzzy set size relationship in possibility theory and derive the possibility factor rotation matrix MM_i according to the definition of the probability measure that represents the size relationship of the fuzzy set.

1) Possibility factor rotation matrix in Zoon *X*:

$$MM_x=\begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix} \tag{24}$$

2) Possibility factor rotation matrix in Zone *a*:

$$MM_1=\begin{bmatrix}\mathrm{POS}(N\ge F_e) & \mathrm{NES}(N>F_e)\\ \mathrm{NES}(N>F_e) & \mathrm{POS}(N\ge F_e)\end{bmatrix} \tag{25}$$

3) Possibility factor rotation matrix in Zone *b*:

$$MM_2=\begin{bmatrix}\mathrm{POS}(F_e\ge N) & \mathrm{NES}(F_e>M)\\ \mathrm{NES}(F_e>M) & \mathrm{POS}(F_e\ge N)\end{bmatrix} \tag{26}$$

4) Possibility factor rotation matrix in Zone *c*:

$$MM_3 = \begin{bmatrix} \mathrm{POS}(F_e \ge M) & \mathrm{NES}(F_e > N) \\ \mathrm{NES}(F_e > N) & \mathrm{POS}(F_e \ge M) \end{bmatrix} \tag{27}$$

5) Possibility factor rotation matrix in Zone *d*:

$$MM_4 = \begin{bmatrix} \mathrm{POS}(M \ge F_e) & \mathrm{NES}(M > F_e) \\ \mathrm{NES}(M > F_e) & \mathrm{POS}(M \ge F_e) \end{bmatrix} \tag{28}$$

6) Possibility factor rotation matrix in Zoon *Y*:

$$MM_y = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \tag{29}$$

The information content I_i in each zone is given by Equation (30). The final possibility oblique factor rotation *MM* is the weighted sum of the information content in each zone, the possibility factor matrix elements in each zone, and Equation (31).

$$I_i = \max_S \mu_{F_e i}(S) \times \log \mu_{F_e i}(S) \ (i = 1,2,3,4) \tag{30}$$

$$MM = \sum_{i=1}^{4} MM_i \times I_i \tag{31}$$

7. Initial and Stopping Condition on Type 2 Possibility Oblique Factor Rotation

In this section, we consider the possibility oblique factor rotation after the initial observation X_0. At first, in zone *X* and zone *Y*, the initial and stopping condition can be derived from the normal Markov process (Takahashi [12]). Second, *N* and F_e is direct sum in zone *a*. Furthermore, *M* and F_e is direct sum in zone *d*. In this case, these rotations are as well as the possibility principal rotations. At last, the initial and stopping conditions in zone *b* and zone *c* are derived from Equation (32) and (33). Note that the type2 fuzzy event *N* can rotate as same as the type2 fuzzy event *M*, because changing *M* to *N* in Equation (32) and (33) is similar to the original version.

(Initial Condition)

1) $(M_{0t}, N_{0t}) = (Z_{10t}, Z_{20t})$ (32)

2) $I_2 \mathrm{POS}((M_t, F_{et}) \ge (Z_{1t}, Z_{2t}) \mid x_0) \le I_3 \mathrm{POS}((M_t X_0, F_{et} X_0) \ge (M_t X_0, N_t X_0))$

3) $I_2 \mathrm{NES}((N_t, F_{et}) \ge (Z_{1t}, Z_{2t}) \mid x_0) \ge I_3 \mathrm{NES}((N_t, F_{et}) \ge (M_t X_0, N_t X_0))$

(Stopping Condition)

1) $(M_{0t}, N_{0t}) = (Z_{10t}, Z_{20t})$ (33)

2) $I_2 \mathrm{POS}((M_t, F_{et}) \ge (Z_{1t}, Z_{2t})) \ge I_3 \mathrm{POS}((M_t, F_{et}) \ge (M_t, N_t))$

3) $I_2 \mathrm{NES}((N_t, F_{et}) \ge (Z_{1t}, Z_{2t})) \le I_3 \mathrm{NES}((N_t, F_{et}) \ge (M_t, N_t))$

The starting and stopping state are obtained in zone b as follows:

(Stating State) $M_{10}(\cdot)_t = M_{20}(\cdot)_t$ (34)

1) $M_{ti0} = F_{ei0} \ (i = 1,2)$

2) $I_2 \mathrm{POS}(M_{ti} \ge F_{eti} \mid X_0) \le I_3 \mathrm{POS}(N_{ti0} \ge F_{eti}) \ (i = 1,2)$

3) $I_2 \mathrm{NES}(M_{ti} \ge F_{eti} \mid X_0) \ge I_3 \mathrm{NES}(N_{ti0} \ge F_{eti}) \ (i = 1,2)$

(Stopping State) $M_{10}(\cdot)_t = M_{20}(\cdot)_t$ (35)

1) $M_{ti0} = F_{ei0}\ (i = 1,2)$

2) $I_2\mathrm{POS}(M_{ti} \ge F_{eti}) \le I_3\mathrm{POS}(N_{ti} \ge F_{eti})(i = 1,2)$

3) $I_2\mathrm{NES}(M_{ti} \ge F_{eti}) \ge I_3\mathrm{NES}(N_{ti} \ge F_{eti})(i = 1,2)$

The starting and stopping state are obtained in zone c as follows:

(Starting State) $N_{10}(\cdot)_t = N_{20}(\cdot)_t$ (36)

1) $N_{ti0} = F_{ei0}\ (i = 1,2)$

2) $I_2\mathrm{POS}(N_{ti} \ge F_{eti} \mid X_0) \le I_3\mathrm{POS}(M_{ti0} \ge F_{eti})(i = 1,2)$

3) $I_2\mathrm{NES}(N_{ti} \ge F_{eti} \mid X_0) \ge I_3\mathrm{NES}(M_{ti0} \ge F_{eti})(i = 1,2)$

(Stopping State) $N_{10}(\cdot)_t = N_{20}(\cdot)_t$ (37)

1) $N_{ti0} = F_{ei0}\ (i = 1,2)$

2) $I_2\mathrm{POS}(N_{ti} \ge N_{eti}) \le I_3\mathrm{POS}(M_{ti} \ge F_{eti})(i = 1,2)$

3) $I_2\mathrm{NES}(N_{ti} \ge N_{eti}) \ge I_3\mathrm{NES}(M_{ti} \ge F_{eti})(i = 1,2)$

8. Approach Forward to Type 2 Possibility Factor Analysis

In this section, we mention the decision making rule for the possibility factor analysis such that can make a decision rotating the possibility factor rotation in the decision making problem. In the every zone shown by the imaging **Figure 1**, we propose the simple decision rule after the possibility factor rotating. In Equation (38), (39) and (40), D_1 and D_2 are the decisions. And $x = U_1(S \mid D_1)$, $y = U_2(S \mid D_2)$, are utility functions. $\mu_N(S)$ is the type 2 membership function in *x*-axis. $\mu_M(S)$ is the type 2 membership function in *y*-axis. And $\pi(S)$ is the possibility prior distribution. Note that the decision maker can obtain these utility functions by the lot method after deciding his type (Risk Aversion, Risk Neutral, Risk Proneness).

1) Zone *x*, *y* (Integral Transfer (maximizing Expected Utility)) (38)

$$E(D_1) = \int \pi(S) \times \mu_N\left(U_1^{-1}(x \mid D_1)\right)\mathrm{d}S$$

$$E(D_2) = \int \pi(S) \times \mu_M\left(U_2^{-1}(y \mid D_2)\right)\mathrm{d}S$$

2) Zone *a*, *d* (max-product method) (39)

$$\pi(D_1) = \max_s \pi(S) \times \mu_N\left(U_1^{-1}(x \mid D_1)\right)$$

$$\pi(D_2) = \max_s \pi(S) \times \mu_M\left(U_2^{-1}(y \mid D_2)\right)$$

3) Zone *b*, *c* (min-max principal) (40)

$$\wedge(D_1) = \min_S \max\left(\pi(S), \mu_N\left(U_1^{-1}(x \mid D_1)\right)\right)$$

$$\wedge(D_2) = \min_S \max\left(\pi(S), \mu_M\left(U_2^{-1}(y \mid D_2)\right)\right)$$

Here, in Equations (38), (39) and (40), after we pick up the bigger measure in the every zone, we can decide the individual optimal decision with these measures. Otherwise, because type 2 fuzzy event *M*, *N* and F_e are the direct sum, we can select the total optimal decision by the max the weighted sum of the information content (30). Note that we may need to obtain the individual optimal decision in the every zone. (Uemura, Inaida [11]).

9. Conclusion

In this paper, we mentioned that Type 2 Fuzzy for no-data problems is derived from simultaneous stochastic differential equations in fuzzy events and is closely related to artificial intelligence in multi-input nonlinear factor analysis. The Type 2 Fuzzy Event is a multidimensional possibility variable error model that simultaneously considers longitudinal and transverse errors introducing the concepts of possibility and necessity, and it is application to main factor analysis based on possibility theory is described. Focusing on the oblique rotation of general factor analysis, the possibility and inevitability measures were defined so that the sum of these measures does not satisfy 1, and the oblique factor rotation matrix was derived by automatically subtracting the sum from 1 for indiscriminate fuzzy events to obtain the possibility and inevitability measures for indiscriminate fuzzy events. If the sum of the weights of the simultaneous stochastic differential equation between the longitudinal and transverse fuzzy variables is greater than 1, no indiscriminate event occurs. For this reason, after normalizing the membership function, the possibility principal factor analysis is applied. Although determinism was not discussed in this paper, it has been proposed that sequential Bayesian inference be employed in cases involving nondiscriminatory events by adding the action of decision withholding (Uemura [1] [4]), because of the information risk of nondiscriminatory events. At last, Japanese call Japanese original traditional decision-making from this world to the other world around another world by the type 2 KIDOU (in Japanese) (Hori [14]). Our proposing theory can apply to deriver the sea wave from the sea wave equation. Specially, when the sea wave equation is regarded as Normal Possibility Process (*i.e.* Gauss Process), we can obtain the sea wave that is Gauss Process (Uemura [9]). Japanese can regard this sea wave process as Sea Goddess/God decision making (Hori [14]).

Conflicts of Interest

The author declares no conflicts of interest regarding the publication of this paper.

References

[1] Uemura, Y. (1991) Decision-Making in Method in Fussy Events. *Journal of the Fussy Society of Japan*, **3**, 123-130. (In Japanese)

[2] Hori Jr., H., Takemura, K. and Matsumoto, Y. (2019) Complex Markov Decision Process. *Journal of Fuzzy Mathematics*, **27**, 957-972.

[3] Okuda, T., Tanaka, H. and Asai, K. (1976) Decision Problems and Information Content in Fuzzy Events. *Journal of the Society of Instrument and Control Engineers*, **12**, 63-68. (In Japanese) https://doi.org/10.9746/sicetr1965.12.63

[4] Uemura, Y. (1990) Fuzzy-Bayes Decision Rule with Indifferent Zone, IIZUKA-90.

[5] Uemura, Y. (1993) A Decision Rule on Fuzzy Events under an Observation. *Journal of Fuzzy Mathematics*, **1**, 39-52.

[6] Zadeh, L.A. (1965) Fuzzy Sets. *Information and Control*, **8**, 338-353. https://doi.org/10.1016/S0019-9958(65)90241-X

[7] Hori Jr., H. (2023) Initial and Stopping Condition in Possibility Principal Factor Analysis. *Journal of Applied Mathematics and Physics*, **11**, 1482-1486. https://doi.org/10.4236/jamp.2023.115097

[8] Hori Jr., H. (2023) Type 2 Fuzzy Stochastic Differential Equations in No-Data Problems and Its Application to Possible Principal Factor Analysis. *Journal of Biomedical Fuzzy Systems Society*, **25**, 65-69. (In Japanese)

[9] Uemura, Y. (2006) Fuzzy Decision Making and Fuzzy Statistics. Fukuro-Shuppan, Fukuyama. (In Japanese)

[10] Dubois, D. and Parade, H. (1988) Possibility Theory. Plenum Press, New York.

[11] Uemura, Y. and Inaida, J. (2023) Possibility State Crimination Method in No-Data Problem. FSS2023, Karuizawa. (In Japanese)

[12] Takahashi, Y. (2011) Kiyoshi Ito's Mathematics. Nippon-Hyoron-Sha, Tokyo. (In Japanese)

[13] Uemura, Y. (1995) The Limit of Using a Probability of a Fuzzy Event in a Fuzzy Decision Problem. *Control and Cybernetics*, **24**, 233-238.

[14] Hori Jr., H. (2022) Type 2 KIDOU and Its Applications. *Annual Conference of the Society for Biomedical Fuzzy Systems*, Himeji, 18 December 2002, 18-20. (In Japanese)

Applied Mathematics, 2023, 14, 684-695
https://www.scirp.org/journal/am
ISSN Online: 2152-7393
ISSN Print: 2152-7385

Construction and Weight Distributions of Binary Linear Codes Based on Deep Holes

Yong Yang, Wenwei Qiu

School of Mathematics and Information, China West Normal University, Nanchong, China
Email: yangyong111752@163.com

How to cite this paper: Yang, Y. and Qiu, W.W. (2023) Construction and Weight Distributions of Binary Linear Codes Based on Deep Holes. *Applied Mathematics*, **14**, 684-695.
https://doi.org/10.4236/am.2023.1410040

Received: September 7, 2023
Accepted: October 9, 2023
Published: October 12, 2023

Abstract

Deep holes are very important in the decoding of generalized RS codes, and deep holes of RS codes have been widely studied, but there are few works on constructing general linear codes based on deep holes. Therefore, we consider constructing binary linear codes by combining deep holes with binary BCH codes. In this article, we consider the 2-error-correcting binary primitive BCH codes and the extended codes to construct new binary linear codes by combining them with deep holes, respectively. Furthermore, three classes of binary linear codes are constructed, and then we determine the parameters and the weight distributions of these new binary linear codes.

Keywords

Linear Codes, MacWilliams Equations, Weight Distribution, Dual Codes, Deep Holes, Covering Radius

1. Introduction

Let $\mathbb{F}_q$ be the finite field with q elements, where q is an odd prime. An $[n,k,d]$ linear code $\mathcal{C}$ is a k-dimensional subspace of the vector space $\mathbb{F}_q^n$ with minimum distance d, where $1 \le k \le n$. The weight of code x is denoted by $wt(x)$. The dual code of $\mathcal{C}$, denoted by $\mathcal{C}^{\perp}$, is defined by

$$\mathcal{C}^{\perp} = \left\{ x \in \mathbb{F}_q^n \mid xc = 0, \text{ for all } c \in \mathcal{C} \right\}.$$

Clearly, $\dim(\mathcal{C}) + \dim(\mathcal{C}^{\perp}) = n$.

For the linear code $\mathcal{C}$ with length n, the number of codewords of weight i denotes $A_i(\mathcal{C})$ with $0 \le i \le n$. The *weight enumerator* of $\mathcal{C}$ is defined by

$$1 + l_1 z + l_2 z^2 + l_3 z^3 + \cdots + l_n z^n,$$

and the sequence $(1, l_1, l_2, \cdots, l_n)$ is said to be the weight distribution of $\mathcal{C}$. If the number of non-zeros in the sequence is t, then we say that the linear code is a t-weight code. The weight distributions of linear codes not only give the important information of linear codes in practice and theory, but also reflect the error-correcting ability of linear codes and the probability of error information occurring during transmission. In general, it is not effortless to determine the weight distributions of linear codes.

The coset of $\mathcal{C}$, denoted by Q, is defined by

$$Q = \{x + c \mid c \in \mathcal{C}\} \subset \mathbb{F}_q^n,$$

where $x \in Q$ is a vector fixed for the given representation. The *weight of Q* is the smallest Hamming weight of the codewords of Q.

For any vector (codeword) $u \in \mathbb{F}_q^n$, the error distance to code $\mathcal{C}$ of a received codeword u is defined by

$$d(u, \mathcal{C}) = \min\{d(u, c) \mid c \in \mathcal{C}\},$$

where the minimum Hamming distance between vectors u and c is defined to be

$$d(u, c) = \left|\{i \mid u_i \neq c_i, 1 \le i \le n\}\right|.$$

The maximum error distance is defined by

$$\rho(\mathcal{C}) = \max\{d(u, \mathcal{C}) \mid u \in \mathbb{F}_q^n\},$$

where ρ is called the covering radius of the linear code $\mathcal{C}$. If the error distance to code $\mathcal{C}$ of a received word $u \in \mathbb{F}_q^n$ reaches the covering radius of linear code $\mathcal{C}$, the vector u is called the deep hole.

Deep holes have been widely studied in RS codes, and the deep holes of standard RS codes are given in [1]. In addition, Zhang *et al.* also gave deep holes for several classes of special codes in [2] [3] [4]. Therefore, most scholars are keen on studying the deep holes of some special codes. However, there is little work on constructing general linear codes from deep holes. Thus, we further consider the use of deep holes to construct some binary linear codes.

Around the 1960s, BCH codes were proposed by Hocquenghem [5] and Bose and Ray-Chaudhuri [6], and the error-correcting codes were studied by Gorenstein and Zierler [7] over finite fields. In 1960, Gorenstein *et al.* [8] showed that the covering radius of binary 2-error-correcting BCH codes was 3, and further, the covering radius of the extended codes was 4. The study of 2-error-correcting BCH codes is very thorough, including covering radius, weight distribution, coset weight distribution and so on. The covering radius of the 2-error-correcting binary primitive BCH codes is known. From the definition of deep holes, we know that the deep holes of these BCH codes exist. Since linear codes play an important role in the fields of data storage, information security and secret sharing, the construction of linear codes is one of the important contents in the current cryptography and coding research. Therefore, it is very meaningful to construct binary linear codes.

In this paper, our main work is to construct some binary linear codes by combining deep holes with BCH codes. Furthermore, we can determine the parameters and the weight distributions of the binary linear codes. Finally, some examples are presented by Magma experiments, which support the weight distributions of these binary linear codes. These experimental results coincide with the theoretical results.

The rest of this paper is outlined as follows. Section 2 states some notations and results about narrow-sense binary primitive BCH codes and linear codes. In Section 3, three classes of binary linear codes are constructed and their parameters are determined. In Section 4, the weight distributions of these binary linear codes are obtained. Finally, the conclusion of this paper is given.

2. Preliminaries

In this section, we state some basic facts and known results about linear codes and narrow-sense binary primitive BCH codes.

2.1. The Weight Distributions of the Linear Code and Its Dual Code

For an $[n,k]$ linear code $\mathcal{C}$ over $\mathbb{F}_q$, and denote its dual by $\mathcal{C}^{\perp}$. The weight distribution of $\mathcal{C}$ can be uniquely determined by the weight distribution of $\mathcal{C}^{\perp}$ and vice versa. This linear relationship is crucial for investigating and calculating weight distribution, and we call it the MacWilliams identity.

Let A_j and $A_j^{\perp}$ be the number of codewords of weight j in $\mathcal{C}$ and $\mathcal{C}^{\perp}$, respectively. The *MacWilliams identity* is defined by

$$\sum_{i=0}^{n-j}\binom{n-i}{j}A_i^{\perp}=q^{k-j}\sum_{i=0}^{j}\binom{n-i}{n-j}A_i,\ \text{ for all } j\in[0,n]. \tag{1}$$

Equivalently, we have

$$\sum_{j=0}^{n} j^r A_j=\sum_{j=0}^{\min\{n,r\}}(-1)^j A_j^{\perp}\left[\sum_{t=j}^{r} t!S(r,t)q^{k-t}(q-1)^{t-j}\binom{n-j}{n-t}\right] \text{ for } r\geq 0, \tag{2}$$

and it is more convenient for us to calculate. But it involves the Stirling numbers $S(r,t)$ of the second kind, where $S(r,t)$ is defined by

$$S(r,t)=\frac{1}{t!}\sum_{i=0}^{t}(-1)^{t-i}\binom{t}{i}i^r \text{ for } r,t\geq 0. \tag{3}$$

In particular, $S(r,t)=0$ if $r<t$ and $S(r,r)=1$.

In binary codes, from (2), we deduce

$$\sum_{j=0}^{n} A_j j^r=\sum_{j=0}^{\min\{n,r\}}(-1)^j A_j^{\perp}\left[\sum_{t=j}^{r}\left(\sum_{i=1}^{t}(-1)^{t-i}\binom{t}{i}i^r\right)2^{k-t}\binom{n-j}{n-t}\right] \text{ for } r\geq 0. \tag{4}$$

From (2), the first four Pless power moments are listed as follows ([9], p. 259):

$$\sum_{j=0}^{n} A_j=q^k;$$

$$\sum_{j=0}^{n} jA_j=q^{k-1}\left(qn-n-A_1^{\perp}\right);$$

$$\sum_{j=0}^{n} j^2A_j=q^{k-2}\left[(q-1)n(qn-n+1)-(2qn-q-2n+2)A_1^{\perp}+2A_2^{\perp}\right];$$

$$\sum_{j=0}^{n} j^3 A_j = q^{k-3}\Big[(q-1)n\left(q^2n^2-2qn^2+3qn-q+n^2-3n+2\right)$$
$$-\left(3q^2n^2-3q^2n-6qn^2+12qn+q^2-6q+3n^2-9n+6\right)A_1^{\perp}$$
$$+6(qn-q-n+2)A_2^{\perp}-6A_3^{\perp}\Big].$$

2.2. Cyclic Codes and Narrow-Sense BCH Codes

An $[n,k,d]$ linear code $\mathcal{C}$ over the finite field $\mathbb{F}_q$, it is said to be cyclic if the codeword $(c_{n-1},c_0,\cdots,c_{n-2})\in\mathcal{C}$ implies $c=(c_0,c_1,\cdots,c_{n-1})\in\mathcal{C}$. For each vector $(c_0,c_1,\cdots,c_{n-1})\in\mathbb{F}_q^n$, define $c_0+c_1x+c_2x^2+\cdots+c_{n-1}x^{n-1}\in\mathbb{F}_q[x]/(x^n-1)$, any code $\mathcal{C}$ corresponds to a subset of quotient ring $\mathbb{F}_q[x]/(x^n-1)$. Note that a linear code $\mathcal{C}$ is cyclic if and only if the corresponding subset is an ideal of the quotient ring $\mathbb{F}_q[x]/(x^n-1)$. Besides, every ideal of $\mathbb{F}_q[x]/(x^n-1)$ is principal. Thus, every code $\mathcal{C}$ can be expressed as $\mathcal{C}=\langle g(x)\rangle$, where $g(x)$ is monic and has the smallest degree. $g(x)$ is said the generator polynomial, and $h(x)=(x^n-1)/g(x)$ is referred to as the check polynomial of $\mathcal{C}$.

Let $n=q^m-1$, where m is an integer with $m>1$. Let α be a primitive element of $\mathbb{F}_{q^m}^*$. In addition, let $m_i(x)$ be the minimal polynomial of α^i with $1\le i\le q^m-2$ over $\mathbb{F}_q$. For any $2\le\eta\le n$, define

$$g_{(q,m,\eta)}(x)=lcm\left(m_1(x),m_2(x),\cdots,m_{\eta-1}(x)\right),$$

where lcm denotes the least common multiple of these minimal polynomials.

Let $\mathcal{C}_{(q,m,\eta)}$ denote the cyclic code with generator polynomial $g_{(q,m,\eta)}(x)$, then we know $\mathcal{C}_{(q,m,\eta)}=\langle g_{(q,m,\eta)}(x)\rangle$. The set $\mathcal{C}_{(q,m,\eta)}$ is described as a narrow-sense primitive BCH code with design distance η. An $[n,k,d]$ linear code $\mathcal{C}$ is said e-error-correcting if $e=\left\lfloor\frac{d-1}{2}\right\rfloor$.

Let E denote the 2-error-correcting binary primitive BCH codes, and denote its dual by $E^{\perp}$. The code E is an $[n=2^m-1,k_E=2^m-2m-1,5]$ linear code, for $m\ge 3$. The code E has parity-check matrix H_E defined by

$$H_E=\begin{bmatrix}1 & \alpha & \alpha^2 & \cdots & \alpha^{n-1}\\ 1 & \alpha^3 & \alpha^6 & \cdots & \alpha^{3(n-1)}\end{bmatrix}.$$

The code E consists of all binary codewords $x=(x_0,x_1,\cdots,x_{n-1})\in\mathbb{F}_2^n$ such that $H_Ex^{\mathrm{T}}=0$. The extended code of B denote $\hat{E}$ and denote its duals by $\hat{E}^{\perp}$. A vector r of $\hat{E}$ is $r=(x_{\infty},x_0,x_1,\cdots,x_{n-1})$ where $x_{\infty}=\sum_{i=0}^{n-1}x_i$. The code $\hat{E}$ has parameters $[N=2^m,k_{\hat{E}}=2^m-2m-1,6]$ for $m\ge 3$. The parity-check matrix $H_{\hat{E}}$ is defined by

$$H_{\hat{E}}=\begin{bmatrix}1 & 1 & 1 & 1 & \cdots & 1\\ 0 & 1 & \alpha & \alpha^2 & \cdots & \alpha^{n-1}\\ 0 & 1 & \alpha^3 & \alpha^6 & \cdots & \alpha^{3(n-1)}\end{bmatrix},$$

where α is an element of order $n=2^m-1$ in $\mathbb{F}_2^n$. The code $\hat{E}$ consists of all binary codewords $r=(x_{\infty},x_0,x_1,\cdots,x_{n-1})\in\mathbb{F}_2^N$ such that $H_{\hat{E}}r^{\mathrm{T}}=0$.

2.3. The Weight Distributions of 2-Error-Correcting Binary Primitive BCH Codes and the Extended Codes

In this subsection, we introduce the weight distributions of the 2-error-correcting binary narrow-sense BCH codes and of their extensions, whose length are respectively 2^m-1 and 2^m.

The code $E^{\perp}$ is an $\left[n=2^m-1, 2m, 2^{m-1}-2^{(m-1)/2}\right]$ linear code in [10], for odd $m \geq 3$. The code $E^{\perp}$ is a binary linear code with the weight distribution in Table 1.

The code $\hat{E}^{\perp}$ is an $\left[N=2^m, 2m+1, 2^{m-1}-2^{m/2}\right]$ linear code in ([11], Table 16.5), for odd $m \geq 3$. The code $\hat{E}^{\perp}$ is a binary linear code with the weight distribution in Table 2.

The code $\hat{E}^{\perp}$ is an $\left[N=2^m, 2m+1, 2^{m-1}-2^{(m-1)/2}\right]$ linear code, for even $m \geq 4$. The code $\hat{E}^{\perp}$ is a binary linear code with the weight distribution in Table 3.

Lemma 1. ([9], *Lemma* 7.5.1) *Let* $\mathcal{C}$ *be an* $[n,k]$ *linear code over* $\mathbb{F}_q$. *Suppose u is a vector in* $\mathbb{F}_q^n$ *but not in* $\mathcal{C}$. *The linear code is generated by* $\mathcal{C}$ *and u, which is an* $[n,k+1]$ *linear code. Let D be this linear code, then we have*

$$A_j(u+C) = A_j(D \setminus C)/(q-1) \text{ for } 0 \leq j \leq n.$$

Table 1. The weight distribution of $E^{\perp}$, for odd $m \geq 3$.

Weights	The number of codewords
0	1
$(n+1)/2-\sqrt{(n+1)/2}$	$n\left((n+1)/4+\sqrt{(n+1)/8}\right)$
$(n+1)/2$	$n\left((n+1)/2+1\right)$
$(n+1)/2+\sqrt{(n+1)/2}$	$n\left((n+1)/4-\sqrt{(n+1)/8}\right)$

Table 2. The weight distribution of $\hat{E}^{\perp}$, for odd $m \geq 3$.

Weights	The number of codewords
0, N	1
$N/2-\sqrt{N/2}$	$N(N-1)/2$
$N/2$	$N(N-1)(N+2)$
$N/2+\sqrt{N/2}$	$N(N-1)/2$

Table 3. The weight distribution of $\hat{E}^{\perp}$, for even $m \geq 4$.

Weights	The number of codewords
0, N	1
$N/2 \pm \sqrt{N}$	$N(N-1)/12$
$N/2$	$(N-1)(N+4)/2$
$N/2 \pm \sqrt{N/4}$	$2N(N-1)/3$

In binary codes, it is clear that we have

$$A_j(u+C)=A_j(D\setminus C) \text{ for } 0\le j\le n.$$

In other words, we have

$$A_j(u+C)=A_j(D)-A_j(C) \text{ for } 0\le j\le n.$$

Theorem 2. ([9], *Th.* 7.3.1) *Let* T *be a set* $\{1,2,3,\cdots,d\}$ *with* $|T|=d$. *The weight distribution of* $\mathcal{C}$ *and* $\mathcal{C}^{\perp}$ *are determined by* $A_1^{\perp},A_2^{\perp},\cdots,A_{d-1}^{\perp}$ *and the* A_i *with* $i\notin T$.

It is very convenient for us to calculate the weight distribution. The following corollary can be deduced from Theorem 2.

Corollary 3. *Let* $\mathcal{C}$ *be an* $[n,k,d]$ *linear code over* $\mathbb{F}_q$, *and denote its dual by* $\mathcal{C}^{\perp}$. *Then the dual code* $\mathcal{C}^{\perp}$ *is a linear code of length n and dimension* $n-k$. *Let T be a set* $\{1,2,3,\cdots,d\}$ *with* $|T|=d$ *and* $A_1=A_2=\cdots=A_{d-1}=0$, *so the weight distribution of* $\mathcal{C}^{\perp}$ *is uniquely determined. If* $r<d$, (4) *is equivalent to*

$$\sum_{j=0}^{n}A_j^{\perp}j^r=\sum_{d=0}^{r}\left(\sum_{i=1}^{d}(-1)^{d-i}\binom{d}{i}i^r\right)2^{n-k-d}\binom{n}{d} \text{ for } 0\le r<d. \tag{5}$$

It is very convenient to calculate the weight distribution of the dual code $\mathcal{C}^{\perp}$. If $r\ge d$, we use (4).

Theorem 4. ([9], *Th.* 1.4.5) *Let* $\mathcal{C}$ *be an* $[n,k,d]$ *linear code over* $\mathbb{F}_q$, *then we have*

- $A_0(\mathcal{C})+A_1(\mathcal{C})+A_2(\mathcal{C})+\cdots+A_n(\mathcal{C})=q^k$.
- $A_0(\mathcal{C})=1$, and $A_1(\mathcal{C})=A_2(\mathcal{C})=\cdots=A_{d-1}(\mathcal{C})=0$.
- If $\mathcal{C}$ is a binary code, which contains the codeword $\mathbf{1}=11\cdots1$. We know $A_j(\mathcal{C})=A_{n-j}(\mathcal{C})$ for $0\le j\le n$.

Thus, for a 2-ary linear code $\mathcal{C}$, if $\mathcal{C}$ contains the codeword $\mathbf{1}=11\cdots1$, then the weight distribution of $\mathcal{C}$ is symmetric.

3. The Parameters of Three Classes of Binary Linear Codes

In this chapter, we construct three classes of binary linear codes based on deep holes, then determine the parameters of these binary linear codes.

Let $\mathcal{C}$ be an $[n,k,d]$ linear code over the finite field $\mathbb{F}_q$, let u be a deep hole of the linear code $\mathcal{C}$, then we construct general linear code $\mathcal{C}_u=\{c+\lambda u\mid c\in\mathcal{C},\lambda\in\mathbb{F}_q\}$. We consider binary BCH codes combined with deep holes to construct general linear codes.

Three classes of binary linear codes are constructed by deep holes combined with the 2-error-correcting binary primitive BCH code and their extended codes, respectively.

Lemma 5. *The code E is an* $[n=2^m-1,k_E=2^m-2m-1,5]$ *linear code. Suppose u is a deep hole of the code E, and construct general binary linear code* $E_u=\{c+\lambda u\mid c\in E,\lambda\in\mathbb{F}_2\}$. *Then the code* E_u *is an* $[n=2^m-1,2^m-2m,3]$ *linear code, and the dual code* $E_u^{\perp}$ *is an* $[n=2^m-1,K_E=2^m-1,2^{m-1}-2^{(m-1)/2}]$

linear code, where m is odd and $m \geq 3$.

Proof. Let $\alpha_1, \alpha_2, \cdots, \alpha_{k_E}$ be a basis for a k_E-dimensional subspace of $\mathbb{F}_q^n$, and the code E is a vector space $span(\alpha_1, \alpha_2, \cdots, \alpha_{k_E})$. Since the covering radius of code E is 3, from the definition of deep hole, the maximum error distance $d(u, E) = 3$. Moreover, the minimum distance of E is 5, it is easy to know that $u \notin E$. The binary linear code E_u is constructed, we have

$$u \in E_u \text{ and } E \subset E_u.$$

Furthermore, we obtain

$$d(u, E) = d(u, E_u) = 3,$$

so the minimum distance of the linear code E_u is 3. The binary code E_u is the vector space $span(\alpha_1, \alpha_2, \cdots, \alpha_{k_E}, u)$. Since $u \notin E$, so the dimension of the linear code E_u is $k_E + 1$. Namely, the code E_u has parameters $[n = 2^m - 1, 2^m - 2m, 3]$.

Let set $S = \{i > 0 \mid A_i(E^\perp) \neq 0\}$ for $0 < i \leq n$. As $E_u^\perp \subset E^\perp$, we get

$$A_i(E_u^\perp) = 0 \text{ for } i \neq 0$$

and $i \notin S$. We denote the minimum Hamming distance of the code $E_u^\perp$ by $d^\perp$, from the weight distribution in Table 1, we have

$$d^\perp = 2^{m-1} - 2^{(m-1)/2} \text{ for odd } m \geq 3.$$

Then the dual code $E_u^\perp$ has parameters $[n = 2^m - 1, K_E = 2m - 1, 2^{m-1} - 2^{(m-1)/2}]$, for odd $m \geq 3$. Therefore, the parameters of the code E_u and of its dual code $E_u^\perp$ are determined separately. □

We similarly construct several classes of binary linear codes and determine their parameters. The proof is similar to that of Lemma 5 and is omitted here. These binary linear codes are as follows.

Lemma 6. *The code* $\hat{E}$ *is an* $[N = 2^m, k_{\hat{E}} = 2^m - 2m - 1, 6]$ *linear code. Suppose u is a deep hole of the code* $\hat{E}$. *We construct general binary linear code* $\hat{E}_u = \{c + \lambda u \mid c \in \hat{E}, \lambda \in \mathbb{F}_2\}$. *Then the code* $\hat{E}_u$ *is an* $[N = 2^m, 2^m - 2m, 4]$ *linear code, and the dual code* $\hat{E}_u^\perp$ *is an* $[N = 2^m, K_{\hat{E}} = 2^m, 2^{m-1} - 2^{(m-1)/2}]$ *linear code, where m is odd and* $m \geq 3$.

Lemma 7. *The code* $\hat{E}$ *is an* $[N = 2^m, k_{\hat{E}} = 2^m - 2m - 1, 6]$ *linear code, suppose* u_1 *is a deep hole of the code* $\hat{E}$. *We construct general binary linear code* $\hat{E}_{u_1} = \{c + \lambda u_1 \mid c \in \hat{E}, \lambda \in \mathbb{F}_2\}$. *Then the code* $\hat{E}_{u_1}$ *has parameters* $[N = 2^m, 2^m - 2m, 4]$, *and the dual code* $\hat{E}_{u_1}^\perp$ *has parameters* $[N = 2^m, K_{\hat{E}} = 2^m, 2^{m-1} - 2^{m/2}]$, *for even* $m \geq 4$.

4. The Weight Distributions of Three Classes of Binary Linear Codes

In this part, we determine the weight distributions of these binary linear codes. From the general linear code $\mathcal{C}_u = \{c + \lambda u \mid c \in \mathcal{C}, \lambda \in \mathbb{F}_q\}$, the weight distributions of these binary linear codes are related to the coset weight distributions of

BCH codes. The coset weight distributions of BCH codes have been studied in the literature [12] [13].

To facilitate the computation of the weight distribution of the dual code, the following lemma can be deduced.

Lemma 8. *Let* $\mathcal{C}$ *be an* $[n,k,d]$ *linear code, and denote its dual by* $\mathcal{C}^{\perp}$. *Then* $A_1(\mathcal{C}) = A_2(\mathcal{C}) = \cdots = A_{d-1}(\mathcal{C}) = 0$ *and* $A_0(\mathcal{C}) = 1$. *Thus by Corollary* 3, *the weight distribution of* $\mathcal{C}^{\perp}$ *can be determined by the first d Pless power moments, we obtain*

$$\begin{cases} \sum_{i=0}^{n} A_i^{\perp} = 2^{n-k}, \\ \sum_{i=0}^{n} iA_i^{\perp} = 2^{n-k-1}n, \\ \sum_{i=0}^{n} i^2 A_i^{\perp} = 2^{n-k-2}n(n+1), \\ \sum_{i=0}^{n} i^3 A_i^{\perp} = 2^{n-k-3}n^2(n+3), \\ \sum_{i=1}^{n} i^4 A_i^{\perp} = 2^{n-k-4}n(n+1)(n^2+5n-2), \\ \sum_{i=1}^{n} i^5 A_i^{\perp} = 2^{n-k-5}n^2(n^3+10n^2+15n-10), \\ \sum_{i=1}^{n} i^6 A_i^{\perp} = 2^{n-k-6}n(n+1)(n^4+14n^3+31n-46n+16), \\ \cdots, \end{cases} \tag{6}$$

where $A_i^{\perp}$ is the number of codewords of weight i in $\mathcal{C}^{\perp}$.

Theorem 9. *The binary code* E_u *is an* $[n = 2^m - 1, 2^m - 2m, 3]$ *linear code, and the dual code* $E_u^{\perp}$ *is an* $\left[n = 2^m - 1, K_E = 2m-1, 2^{m-1} - 2^{(m-1)/2}\right]$ *linear code, where m is odd and* $m \geq 3$. *Moreover, the weight distribution of* $E_u^{\perp}$ *is shown in* **Table 4**.

Proof. Let $U = \{w_i > 0 \mid A_{w_i}(E^{\perp}) \neq 0\}$, as $E_u^{\perp} \subset E^{\perp}$, we have $A_{w_i}(E_u^{\perp}) = 0$ if $w_i \neq 0$ and $w_i \notin U$. From **Table 1**, it is easily seen that the minimum weight distribution of $E_u^{\perp}$ is at least 3. Thus, there are three nonzero weights of $E_u^{\perp}$ as follows:

$$w_1 = (n+1)/2 - \sqrt{(n+1)/2},\ w_2 = (n+1)/2,\ w_3 = (n+1)/2 + \sqrt{(n+1)/2}.$$

So the weight of the code $E_u^{\perp}$ contains $\{w_1, w_2, w_3\}$. Let $A_{w_i}^{\perp}$ be the total number of codewords with weight w_i in $E_u^{\perp}$. In addition, let $A_i = A_i(E_u)$ and $A_i^{\perp} = A_i(E_u^{\perp})$, where $0 \leq i \leq n$. For binary linear codes, then the first three Pless power moments from Lemma 8, we obtain

$$\begin{cases} \sum_{j=1}^{3} A_{w_j}^{\perp} = 2^{K_E} - 1, \\ \sum_{j=1}^{3} w_j A_{w_j}^{\perp} = 2^{K_E - 1}n, \\ \sum_{j=1}^{3} w_j^2 A_{w_j}^{\perp} = 2^{K_E - 2}n(n+1), \end{cases} \tag{7}$$

where $2^{K_E} = 2^{2m-1}$ and $n = 2^m - 1$.

By solving this system of equations, we obtain the results in **Table 4**. The proof is complete. □

Two examples are presented by Magma experiments, which support Theorem 9.

Table 4. The weight distribution of $E_u^{\perp}$, for odd $m \geq 3$.

Weights	The number of codewords
0	1
$(n+1)/2-\sqrt{(n+1)/2}$	$(n-1)\left((n+1)/2+\sqrt{(n+1)/2}\right)/4$
$(n+1)/2+\sqrt{(n+1)/2}$	$(n-1)\left((n+1)/2-\sqrt{(n+1)/2}\right)/4$
$(n+1)/2$	$(3n^2+8n-3)/8$

Example 1. *Let* $m=3$ *and let the deep hole vector* $u=(1,1,1,0,\cdots,0)$, *the binary code* E_u *has parameters* $[7,2,3]$. *In Theorem* 9, *the code* $E_u^{\perp}$ *is a* $[7,5,2]$ *binary linear code with the weight enumerator* $1+9z^2+19z^4+3z^6$.

Example 2. *Let* $m=5$ *and let the deep hole vector* $u=(1,1,1,0,\cdots,0)$, *the binary linear code* E_u *has parameters* $[31,22,3]$. *In Theorem* 9, *the code* $E_u^{\perp}$ *is a* $[31,9,12]$ *binary linear code with the weight enumerator* $1+150z^{12}+271z^{16}+90z^{20}$.

Theorem 10. *The binary code* $\hat{E}_u$ *is an* $\left[N=2^m, 2^m-2m, 4\right]$ *linear code, and the dual code* $\hat{E}_u^{\perp}$ *is an* $\left[N=2^m, K_{\hat{E}}=2m, 2^{m-1}-2^{(m-1)/2}\right]$ *linear code, where m is odd and* $m \geq 3$. *Moreover, the weight distribution of* $\hat{E}_u^{\perp}$ *is shown in* Table 5.

Proof. Let $U_1=\left\{w_j>0 \mid A_{w_j}\left(\hat{E}^{\perp}\right)\neq 0\right\}$, as $\hat{E}_u^{\perp}\subset\hat{E}^{\perp}$, we have $A_{w_j}\left(\hat{E}_u^{\perp}\right)=0$ if $w_j\neq 0$ and $w_j\notin U_1$. From Table 2, it is easily seen that the minimum weight distribution of $\hat{E}_u^{\perp}$ is at least 4. Therefore, we know that the code $\hat{E}_u^{\perp}$ has the following four nonzero weights:

$$w_1=N/2-\sqrt{N/2},\ w_2=N/2,\ w_3=N/2+\sqrt{N/2},\ w_4=N.$$

The weight of the code $\hat{E}_u^{\perp}$ contains $\{w_1,w_2,w_3,w_4\}$. Let $A_{w_j}^{\perp}$ be the total number of codewords with Hamming weight w_j in $\hat{E}_u^{\perp}$, and let $A_i=A_i\left(\hat{E}_u\right)$ and $A_i^{\perp}=A_i\left(\hat{E}_u^{\perp}\right)$ for $0\leq i\leq N$. Since $\hat{E}$ is even and contains the codeword $1=111\cdots1$. From Theorem 4, we have $A_{w_j}^{\perp}=A_{N-w_j}^{\perp}$ and $A_0^{\perp}=A_N^{\perp}=1$ for $w_j\in U_1$. For the binary linear code, then the first and the third Pless power moments from Lemma 8, we obtain

$$\begin{cases}\sum_{j=1}^{4}A_{w_j}^{\perp}=2^{K_{\hat{E}}}-1,\\ \sum_{j=1}^{4}w_j^2A_{w_j}^{\perp}=2^{K_{\hat{E}}-2}N(N+1),\end{cases} \tag{8}$$

where $2^{K_{\hat{E}}}=2^{2m}$ and $N=2^m$.

By solving this system of equations, we obtain the results in Table 5. The proof is complete. □

Two examples are presented by Magma experiments, which support Theorem 10.

Example 3. *Let* $m=3$ *and let the deep hole vector* $u=(1,1,1,1,0,\cdots,0)$, *the binary linear code* $\hat{E}_u$ *has parameters* $[8,2,4]$. *In Theorem* 10, *the code* $\hat{E}_u^{\perp}$ *is a* $[8,6,2]$ *binary linear code with the weight enumerator* $1+12z^2+38z^4+12z^6+z^8$.

Table 5. The weight distribution of $\hat{E}_u^{\perp}$, for odd $m \geq 3$.

Weights	The number of codewords
0, N	1
$N/2-\sqrt{N/2}$	$N^2/4-N/2$
$N/2+\sqrt{N/2}$	$N^2/4-N/2$
$N/2$	$N^2/2+N-2$

Example 4 *Let* $m=5$ *and let the deep hole vector* $u=(1,1,1,0,1,0,\cdots,0)$, *the binary linear code* $\hat{E}_u$ *has parameters* $[32,22,4]$. *In Theorem* 10, *the code* $\hat{E}_u^{\perp}$ *is a* $[32,10,12]$ *binary linear code with the weight enumerator* $1+240z^{12}+542z^{16}+240z^{20}+z^{32}$.

Theorem 11. *The binary code* $\hat{E}_{u_1}$ *is an* $\left[N=2^m,2^m-2m,4\right]$ *linear code, and the dual code* $\hat{E}_{u_1}^{\perp}$ *is an* $\left[N=2^m,K_{\hat{E}}=2m,2^{m-1}-2^{m/2}\right]$, *where m is even and* $m\geq 4$. *Moreover, the weight distribution of* $\hat{E}_{u_1}^{\perp}$ *is shown in* **Table 6**.

Proof. Let $U_2=\left\{w_i>0 \mid A_{w_i}\left(\hat{E}^{\perp}\right)\neq 0\right\}$, as $\hat{E}_{u_1}^{\perp}\subset\hat{E}^{\perp}$, we have $A_{w_i}\left(\hat{E}_{u_1}^{\perp}\right)=0$ if $w_i\neq 0$ and $w_i\notin U_2$. From **Table 3**, it is easily seen that the minimum weight distribution of $\hat{E}_{u_1}^{\perp}$ is at least 6. Thus, there are six nonzero weights of $\hat{E}_{u_1}^{\perp}$ as follows:

$$w_1=N/2-\sqrt{N},\ w_2=N/2-\sqrt{N/4},\ w_3=N/2,$$
$$w_4=N/2+\sqrt{N/4},\ w_5=N/2-\sqrt{N},\ w_6=N.$$

The weight of the code $\hat{E}_{u_1}^{\perp}$ contains $\{w_1,w_2,w_3,w_4,w_5,w_6\}$. Let $A_{w_i}^{\perp}$ be the total number of codewords with Hamming weight w_i in $\hat{E}_{u_1}^{\perp}$, and let $A_i=A_i\left(\hat{E}_{u_1}\right)$ and $A_i^{\perp}=A_i\left(\hat{E}_{u_1}^{\perp}\right)$ for $0\leq i\leq N$. Since $\hat{E}$ is even and contains the code word $1=111\cdots 1$. From theorem 4, we know that $A_{w_j}^{\perp}=A_{N-w_j}^{\perp}$ and $A_0^{\perp}=A_N^{\perp}=1$ for $w_j\in U_2$. For this binary linear code, then the first, the third and the fifth Pless power moments from Lemma 8 and Corollary 3, we obtain

$$\begin{cases}\sum_{j=1}^{6}A_{w_j}^{\perp}=2^{K_{\hat{E}}}-1,\\ \sum_{j=1}^{6}w_j^2A_{w_j}^{\perp}=2^{K_{\hat{E}}-2}N(N+1),\\ \sum_{j=1}^{6}w_j^4A_{w_j}^{\perp}=2^{K_{\hat{E}}-4}\left(N(N+1)\left(N^2+5N-2\right)+24A_4\right),\end{cases}\tag{9}$$

where $2^{K_{\hat{E}}}=2^{2m}$ and $N=2^m$. Since $\hat{E}_{u_1}$ has parameters $\left[N=2^m,2^m-2m,4\right]$, the minimum Hamming distance is 4, so $A_1=A_2=A_3=0$. We need to determine the number of codewords of weight 4 in $\hat{E}_{u_1}$. Since u_1 is a vector in $\mathbb{F}_2^N$ but not in $\hat{E}$. From Lemma 1, we have

$$A_4\left(\hat{E}_{u_1}\right)=A_4\left(\hat{E}\right)+A_4\left(\hat{E}+u_1\right).$$

The minimum distance of $\hat{E}$ is 6, thus we have $A_4\left(\hat{E}\right)=0$, so

$$A_4\left(\hat{E}_{u_1}\right)=A_4\left(\hat{E}+u_1\right).$$

The number of codewords of weight 4 in the coset $\hat{E}+u_1$ is determined in the literature ([13], Remark 4), then we obtain

Table 6. The weight distribution of $\hat{E}_{u_1}^{\perp}$, for even $m \geq 4$.

Weights	The number of codewords
$0, N$	1
$N/2 \pm \sqrt{N}$	$(N^2 - 4N)/24$
$N/2$	$N^2/4 + N - 2$
$N/2 \pm \sqrt{N/4}$	$(N^2 - N)/3$

$$A_4(\hat{E} + u_1) = N(N-4)/24.$$

Thus, we have

$$A_4(\hat{E}_{u_1}) = N(N-4)/24.$$

By solving this system of equations, we obtain the results in Table 6. The proof is complete. □

Two examples are presented by Magma experiments, which support Theorem 11.

Example 5. *Let* $m = 4$ *and let the deep hole vector* $u_1 = (1,1,1,0,1,0,\cdots,0)$, *the binary linear code* $\hat{E}_{u_1}$ *has parameters* $[16,8,4]$. *In Theorem* 11, *the code* $\hat{E}_{u_1}^{\perp}$ *is a* $[16,8,4]$ *binary linear code with the weight enumerator* $1 + 8z^4 + 80z^6 + 78z^8 + 80z^{10} + 8z^{12} + z^{16}$.

Example 6. *Let* $m = 6$ *and let the deep hole vector* $u_1 = (1,1,1,0,1,0,\cdots,0)$, *the binary linear code* $\hat{E}_{u_1}$ *has parameters* $[64,48,4]$. *In Theorem* 11, *the code* $\hat{E}_{u_1}^{\perp}$ *is a* $[64,12,24]$ *binary linear code with the weight enumerator* $1 + 160z^{24} + 1344z^{28} + 1086z^{32} + 1344z^{36} + 160z^{40} + z^{64}$.

5. Concluding Remarks

In this paper, we consider binary 2-error-correcting BCH codes combined with deep holes to construct general linear codes $\mathcal{C}_u = \{c + \lambda u \mid c \in \mathcal{C}, \lambda \in \mathbb{F}_2\}$, where u is a deep hole of the codes $\mathcal{C}$. Therefore, we not only construct three classes of binary linear codes, but also determine the parameters and the weight distributions of these binary linear codes. Furthermore, we wish to construct more general linear codes related to deep holes.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

[1] Wu, R.J. and Hong, S.F. (2012) On Deep Holes of Standard Reed-Solomon Codes. *Science China Mathematics*, **55**, 2447-2455. https://doi.org/10.1007/s11425-012-4499-3

[2] Zhang, J., Fu, F.W. and Liao, Q.Y. (2013) Deep Holes of Generalized Reed-Solomon Codes. *Scientia Sinica Mathematica*, **43**, 727-740. (In Chinese)

https://doi.org/10.1360/012012-30

[3] Zhang, J., Wan, D.Q. and Kaipa, K. (2019) Deep Holes of Projective Reed-Solomon Codes. *IEEE Transactions on Information Theory*, **66**, 2392-2401. https://doi.org/10.1109/TIT.2019.2940962

[4] Zhang, J. and Wan, D.Q. (2023) On Deep Holes of Elliptic Curve Codes. *IEEE Transactions on Information Theory*, **69**, 4498-4506. https://doi.org/10.1109/TIT.2023.3257320

[5] Hocquenghem, A. (1959) Codes correcteurs d'rreurs. *Chiffres* (*Paris*), **2**, 147-156.

[6] Bose, R.C. and Ray-Chaudhuri, D.K. (1960) On a Class of Error Correcting Binary Group Codes. *Information and Control*, **3**, 68-79. https://doi.org/10.1016/S0019-9958(60)90287-4

[7] Gorenstein, D. and Zierler, N. (1961) A Class of Error-Correcting Codes in p^m Symbols. *Journal of the Society for Industrial and Applied Mathematics*, **9**, 207-214. https://doi.org/10.1137/0109020

[8] Gorenstein, D., Peterson, W.W. and Zierler, N. (1960) Two-Error Correcting Bose-Chaudhuri Codes Are Quasi-Perfect. *Information and Control*, **3**, 291-294. https://doi.org/10.1016/S0019-9958(60)90877-9

[9] Huffman, W.C. and Pless, V. (2003) Fundamentals of Error-Correcting Codes. Cambridge University Press, Cambridge. https://doi.org/10.1017/CBO9780511807077

[10] MacWilliams, F.J. and Sloane, N.J.A. (1977) The Theory of Error-Correcting Codes (I and II). North-Holland Publishing Company, Amsterdam.

[11] Berlekamp, E.R. (1968) Algebraic Coding Theory. McGraw-Hill, New York.

[12] Assmus, E. and Mattson, H. (1978) The Weight-Distribution of a Coset of a Linear Code (Corresp.). *IEEE Transactions on Information Theory*, **24**, 497-497. https://doi.org/10.1109/TIT.1978.1055903

[13] Charpin, P. (1994) Weight Distributions of Cosets of Two-Error-Correcting Binary BCH Codes, Extended or Not. *IEEE Transactions on Information Theory*, **40**, 1425-1442. https://doi.org/10.1109/18.333859

Scientific Research Publishing

Applied Mathematics, 2023, 14, 696-703
https://www.scirp.org/journal/am
ISSN Online: 2152-7393
ISSN Print: 2152-7385

Probability Theory Predicts That Winning Streak Is a Shortcut for the Underdog Team to Win the World Series

Motohisa Osaka

Department of Basic Science, Nippon Veterinary and Life Science University, Tokyo, Japan
Email: osaka@nms.ac.jp

How to cite this paper: Osaka, M. (2023) Probability Theory Predicts That Winning Streak Is a Shortcut for the Underdog Team to Win the World Series. *Applied Mathematics*, **14**, 696-703.
https://doi.org/10.4236/am.2023.1410041

Received: September 18, 2023
Accepted: October 13, 2023
Published: October 16, 2023

Abstract

It is common for two teams or two players to play a game in which the first one to win a majority of the initially determined number of matches wins the championship. We will explore the probabilistic conditions under which a team (or player) that is considered weak may win the championship over a team (or player) that is considered strong, or a game may go all the way to the end, creating excitement among fans. It is unlikely to occur if the initially estimated probability remains constant when the weaker one wins each game against the stronger one. The purpose of this study is to identify probabilistically what conditions are necessary to increase the probability of such an outcome. We examine probabilistically by quantifying momentum gains to see if momentum gains by a weaker team (or player) winning a series of games would increase the likelihood of such an outcome occurring. If the weaker one gains momentum by winning a series of games and the probability of winning the next game is greater than the initial probability, we can see that such a result will occur in this study. Especially when the number of games is limited to seven, the initial probability that a weaker one will beat a stronger one in each game must be 0.35 or higher in order to win the championship and excite the fans by having the game go all the way to the end.

Keywords

Game, Sports, Underdog, World Series, Upset Championship

1. Introduction

Baseball is a very popular sport in the United States. In particular, the World Series, which determines the best team of the year, is the most exciting event of the year. The team that wins four games first wins the championship. According to Nahin, the World Series has been decided in an average of 5.8 games under this current rule [1]. The most exciting pattern of this event is when a team with a weak record wins the championship, or (more likely) when the game goes down to the last game. This can be generalized not only to baseball, but also to other sports, such as when two winning teams play each other and the team that wins the majority of the games decided first is declared the winner. The probability that the weaker team will win each game against the stronger team will be less than 0.5, but it is clear that the greater the number of games played, the further away from the championship. So what conditions are necessary for a team that is considered weak to increase its chances of winning the championship? The first condition is that the probability that a team considered weak will beat a strong team is not constant, but varies depending on the situation. This could happen, for example, when a weaker team beats a stronger team once and gains momentum to show unexpected strength in the next match. In this case, it is assumed that the probability of the weaker team beating the stronger team is several times greater than the initial probability. Even such a probability theory problem that cannot be formulated can be programmed and solved by numerical computation [2]-[8].

The two objectives of this study are: 1) to determine the number of games in which the weaker team (or player) has the highest probability of winning the majority of all games when the probability of the weaker one beating the stronger one is constant, and 2) to determine the relationship between the probability of the weaker one winning the championship and the number of games in which it wins the majority of the games, when the initial probability p is A (>1) times larger in case, it wins a series of games.

2. Methods

To calculate the probability P of the weaker team W (a team with a weak record is hereafter referred to as the weaker team) winning the championship, the details are set as follows. The probability p of the weak team is assumed to be in the range $0 < p < 0.5$. The maximum number of games N is assumed to be an odd number, and the team that wins the majority of these games, *i.e.* $(n + 1)/2$ games, first, is assumed to be the winner. The game ends at the moment the winning team is determined, even if the maximum number of games has not been reached.

First, calculate P when p is constant. When W is determined to win the championship, the winner of the last match is W.

$$P = \sum_{n=M}^{N} \binom{n}{m} p^{M-1} (1-p)^{n-M} p. \text{ Then } M \equiv (N+1)/2 \tag{1}$$

P was computed with $N = 3, 5, 7, \ldots, 11$, and $p = 0.1, 0.15, 0.25, \ldots, 0.4$.

Next, calculate P when p varies with the situation. Assume that p is A (>1) times larger when W wins a series of games. A team with a strong record is hereafter denoted as the strong team S.

The following specific example shows how to calculate the probability that W will win. Example 1) Suppose $N = 7$ and the final result of winning or losing the game is SWWWSW: the probability of W winning in the second game is p, the probability of winning in the third game is $p \times A$, the probability of winning in the fourth game is $p \times A \times A$, and the probability of winning in the sixth game is equal to $p \times A \times A$ in the fourth game since the fifth game was lost. $p \times A \times A$ is 0.95 if $p \times A \times A$ is greater than 1. Hence the probability of SWWWSW occurring is $(1 - p) \times p \times (p \times A) \times (p \times A \times A) \times (1 - p \times A \times A) \times (p \times A \times A)$.

Example 2) Suppose $N = 7$ and the final result of winning or losing the game is WWWSSW: the probability of W winning in the second game is $p \times A$, and the probability of winning in the third game is $p \times A \times A$. In this case, if $p \times A \times A \geq 1$, $p \times A \times A$ is replaced by 0.95. The probability of losing in the fourth game and the fifth game is 0.05, and the probability of winning in the sixth game is 0.95. Hence the probability of WWWSSW occurring is $p \times (p \times A) \times 0.95 \times 0.05 \times 0.05 \times 0.95$.

Finally, P is computed with $N = 3, 5, 7, \ldots, 11$, $p = 0.1, 0.15, 0.25, \ldots, 0.4$, and $A = 1.25, 1.5, 2$. P was obtained by programming and numerical computation.

3. Results

1) A case of $A = 1$ corresponds to the case where p does not depend on the outcome of every match. The probability of W winning always decreases as the number of games increases (**Figure 1**). Moreover, P is always less than p.

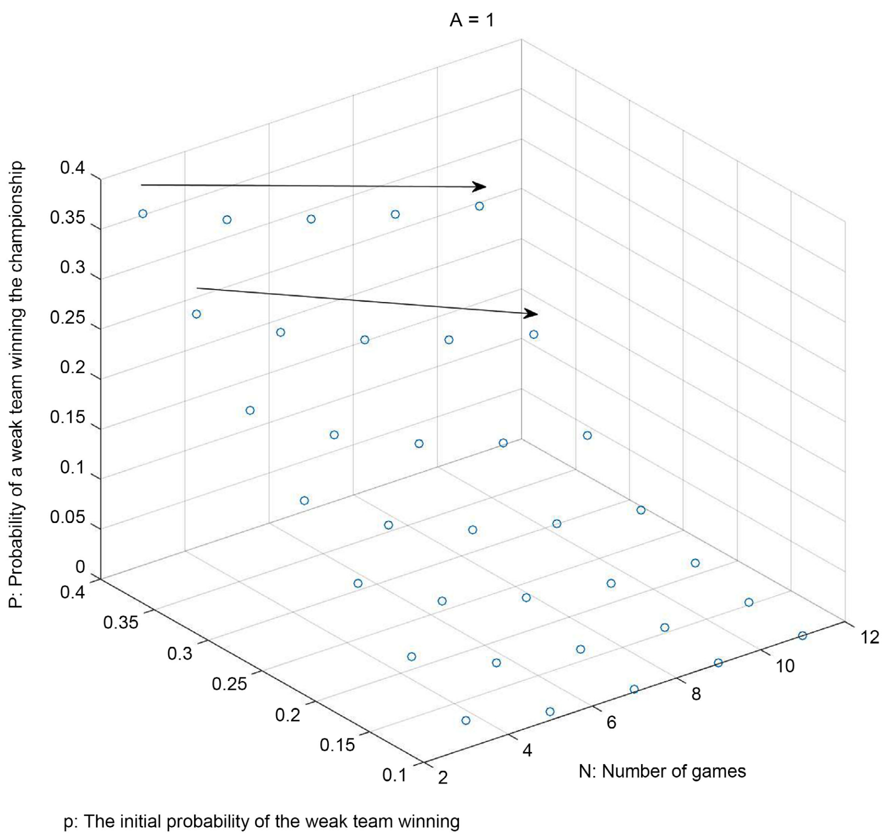


Figure 1. 3D plot in case $A = 1$.

2) The following results are for the case where p increases A-fold when W wins consecutively.

a) A case of $A = 1.25$ (**Figure 2**)

When $p \leq 0.35$, P decreases as the number of matches N increases; when $p = 0.4$, P is almost constant, and $P < p$.

b) A case of $A = 1.5$ (**Figure 3**)

When $p \leq 0.25$, P decreases as the number of matches N increases, and $P < p$. When $p = 0.3$, P is almost constant. When $p \geq 0.35$, P increases as the number of matches N increases, and $P > p$. Especially, when $p = 0.4$ and $N = 7$, $P = 0.475$.

c) A case of $A = 2$ (**Figure 4** & **Figure 5**)

When $p \leq 0.15$, P decreases as the number of matches N increases, and $P < p$. When $p = 0.25$, P is almost constant. When $p \geq 0.3$, P increases as the number of matches N increases, and $P > p$. Especially, when $p = 0.35$ and $N = 7$, $P = 0.437$, and when $p = 0.4$ and $N = 7$, $P = 0.4833$.

4. Discussion

The rest of the public as well as baseball fans will be excited during the season when a weaker team W wins the World Series against a stronger team S, or when the game goes down to the last minute with no one knowing which way the game will go. Such exciting championship games are not limited to baseball. For

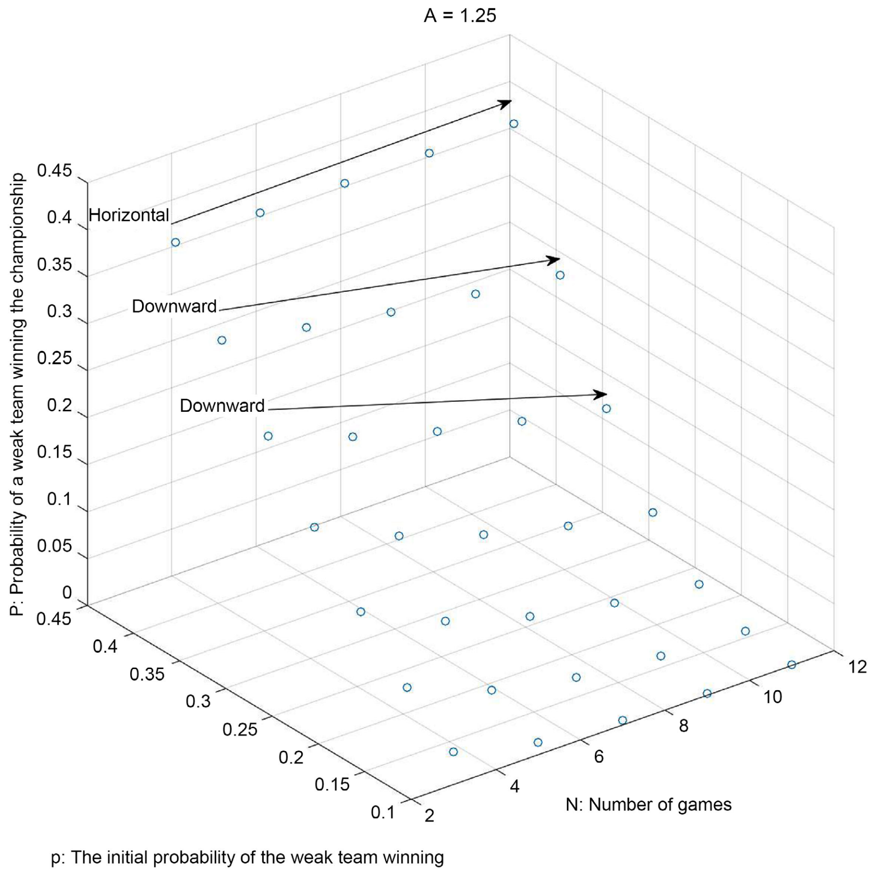


Figure 2. 3D plot in case $A = 1.25$.

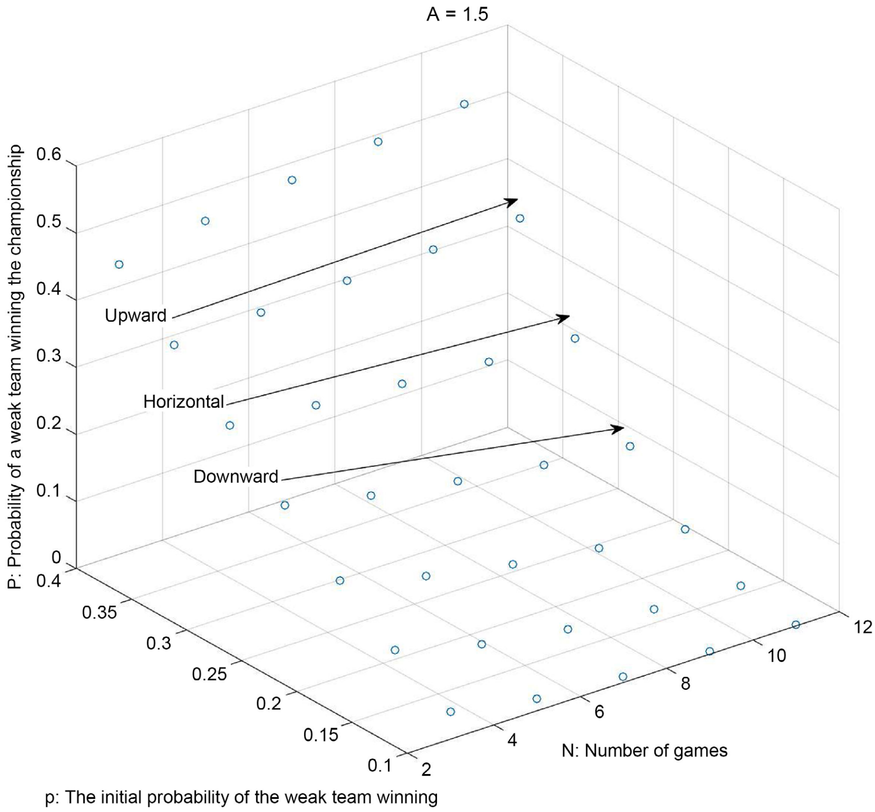


Figure 3. 3D plot in case $A = 1.5$.

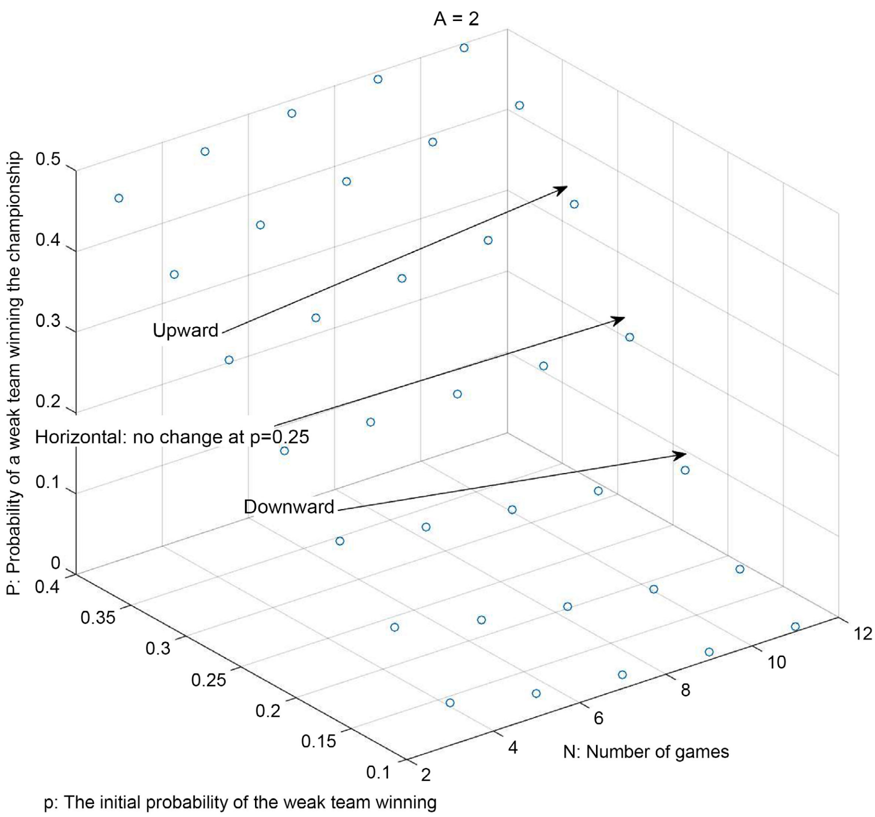


Figure 4. 3D plot in case $A = 2$.

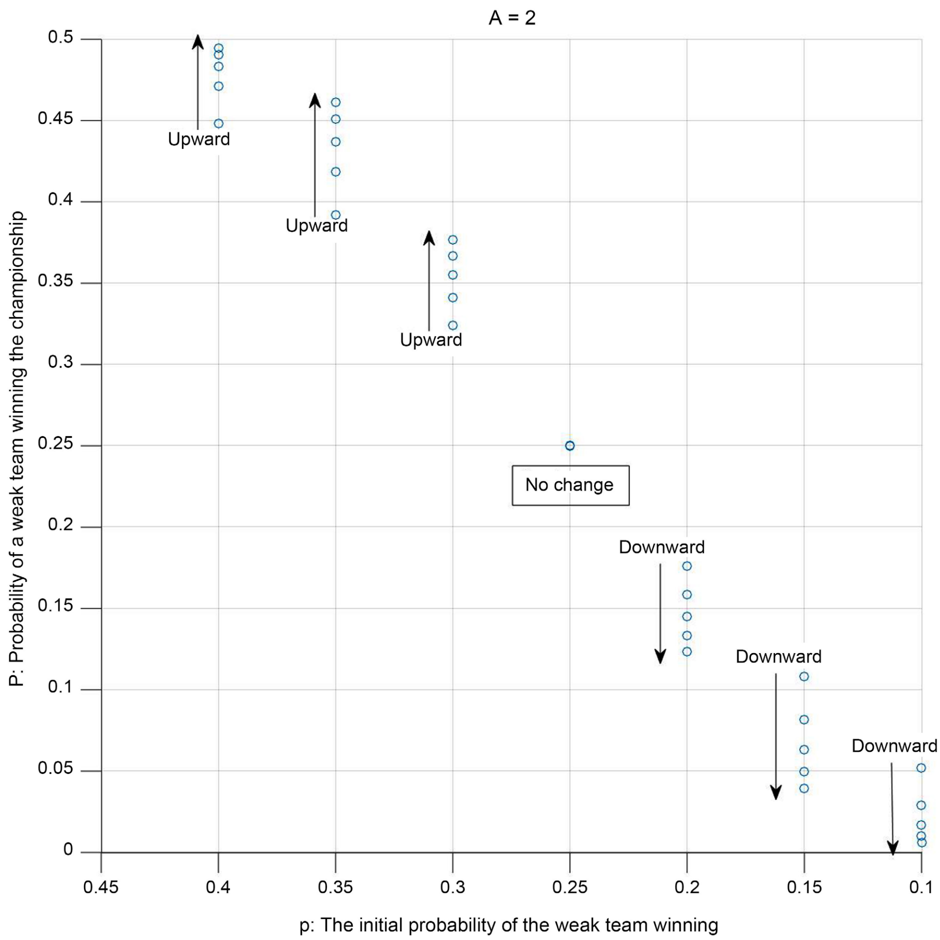


Figure 5. 2D plot in case $A = 2$.

example, in the championship game of the Japanese chess title, the first four winners out of seven games are awarded the title. When p does not depend on the outcome of every match, however, we find that such things are much less likely to happen. Moreover, as the number of games N increases, it becomes more and more difficult for such a thing to happen. Therefore, the answer to the first objective of this study is that if the probability p per game of the weaker team W is constant, the team with a smaller number of games N initially determined still has a chance to win the championship.

The answer to the second objective of this study is as follows: 1) when $A = 1.25$ and $p \leq 0.35$, the probability P of W winning decreases with the number of games regardless of the value of p; 2) when $A = 1.5$ and $p \geq 0.35$, P increases with the number of games, and $P > p$; 3) when $A = 2$ and $p \geq 0.3$, P increases with the number of games, and $P > p$, and 4) when $A = 1.5$, $p = 0.4$, and $N = 7$, $P = 0.475$, and when $A = 2$, $p = 0.35$, and $N = 7$, $P = 0.437$. Hence, even if $p = 0.35$, we would expect W to win or have a close and exciting game. If $p < 0.3$, it will still be difficult to win the championship even if the winning streak gains momentum.

The question is how to interpret this A in the real world. If the supposedly weaker team (or player) wins again, it will probably gain momentum and have a greater chance of beating the supposedly stronger team in the next game, or the supposedly stronger team (or player) will lose confidence and have a greater chance of losing in the next game. In this study, the degree to which the weaker team (or player) gains momentum is expressed as A. The number of games initially determined for the World Series and the Japanese chess title decisive matches is $N = 7$, and the results of this study can be interpreted as follows: comparing the cases $p = 0.35$ and $p = 0.4$, a consecutive win at $p = 0.35$ would give the team more momentum than at $p = 0.4$. As a result, the value of P when $p = 0.35$ and $A = 2$ is almost comparable to the value of P when $p = 0.4$ and $A = 1.5$. In other words, these results suggest that the lower the initial probability of a weaker team (or player) to win, the more momentum it will gain when it wins again against a stronger team (or player). From the opposite perspective, a stronger team (or player) will be further away from the championship if it loses a series of games. The difficulty in quantifying the degree to which the weaker team (or player) gains momentum is a limitation of this study.

5. Conclusion

The results of this study apply to all cases where two teams (or players) play multiple games and the first one to win the majority of the determined number of games wins the championship. In order for a weaker team (or player) to win a majority of the games against a stronger team (or player), or for a game to go all the way to the end, the shortcut is to increase the probability of winning by winning a series of games. Especially when the number of games is limited to seven, the initial probability that a weaker team (or player) will beat a stronger team (or player) in each game must be 0.35 or higher in order to win the championship and excite the fans by having the game go all the way to the end.

Conflicts of Interest

The author declares no conflicts of interest regarding the publication of this paper.

References

[1] Nahin, P.J. (2000) Duelling Idiots and Other Probability Puzzlers. Princeton University Press, Princeton.

[2] Osaka, M. (2014) Probability Theory Predicts That Chunking into Groups of Three or Four Items Increases the Short-Term Memory Capacity. *Applied Mathematics*, **5**, 1474-1484. https://doi.org/10.4236/am.2014.510140

[3] Osaka, M. (2017) Probability Theory Predicts That Group Survival May Be Guaranteed for Groups with More Than 10 Elements. *Applied Mathematics*, **8**, 1745-1760. https://doi.org/10.4236/am.2017.812125

[4] Osaka, M. (2017) Modified Kuramoto Phase Model for Simulating Cardiac Pacemaker Cell Synchronization. *Applied Mathematics*, **8**, 1227-1238. https://doi.org/10.4236/am.2017.89092

[5] Osaka, M. (2019) A Probabilistic Method to Determine Whether the Speed of Light Is Constant. *Applied Mathematics*, **10**, 51-59. https://doi.org/10.4236/am.2019.102005

[6] Osaka, M. (2019) A Mathematical Model Reveals That Both Randomness and Periodicity Are Essential for Sustainable Fluctuations in Stock Prices. *Applied Mathematics*, **10**, 383-396. https://doi.org/10.4236/am.2019.106028

[7] Osaka, M. (2021) A Modified Right Helicoid Can Simulate the Inner Structure of the Cochlea in the Hearing Organ of Mammals. *Applied Mathematics*, **12**, 399-406. https://doi.org/10.4236/am.2021.125028

[8] Osaka, M. (2022) Probability of Matching All Types of Prizes for the First Time Is Maximized at a Surprisingly Early Number of Trials. *Applied Mathematics*, **13**, 869-877. https://doi.org/10.4236/am.2022.1311055

Scientific Research Publishing

Applied Mathematics, 2023, 14, 704-717
https://www.scirp.org/journal/am
ISSN Online: 2152-7393
ISSN Print: 2152-7385

Constructing Confidence Regions for Autoregressive-Model Parameters

Jan Vrbik

Department of Mathematics and Statistics, Brock University, St. Catharines, Canada
Email: jvrbik@brocku.ca

How to cite this paper: Vrbik, J. (2023) Constructing Confidence Regions for Autoregressive-Model Parameters. *Applied Mathematics*, **14**, 704-717.
https://doi.org/10.4236/am.2023.1410042

Received: August 28, 2023
Accepted: October 28, 2023
Published: October 31, 2023

Abstract

We discuss formulas and techniques for finding maximum-likelihood estimators of parameters of autoregressive (with particular emphasis on Markov and Yule) models, computing their asymptotic variance-covariance matrix and displaying the resulting confidence regions; Monte Carlo simulation is then used to establish the accuracy of the corresponding level of confidence. The results indicate that a direct application of the Central Limit Theorem yields errors too large to be acceptable; instead, we recommend using a technique based directly on the natural logarithm of the likelihood function, verifying its substantially higher accuracy. Our study is then extended to the case of estimating only a subset of a model's parameters, when the remaining ones (called nuisance) are of no interest to us.

Keywords

Markov, Yule and Autoregressive Models, Maximum Likelihood Function, Asymptotic Variance-Covariance Matrix, Confidence Intervals, Nuisance Parameters

1. Introduction

The m^{th}-order **autoregressive** models (see [1]) are flexible enough to describe a large assortment of *stationary* time-series data, thus enabling us to make reasonably reliable predictions of future observations. This rests on our ability to accurately estimate parameters of these models, based on a given set of past observations (see [2]); we also need to establish the model's smallest *order* capable of achieving adequate agreement with available data.

We start by reviewing basic formulas for constructing the **likelihood function** (*LF*) of any such model, whose maximization then results in maximum-likelihood (ML) *estimates* (these are the actual numerical values thus obtained) of the model's parameters (note that the same MLEs, seen as *random variables*, *i.e.* functions of yet-to-be taken observations, are called *estimators*). We then proceed with an asymptotic theory of the estimators' sampling distribution (see [3]), and a way of constructing an accurate confidence interval (region) for one (or more) of the parameters; this requires abandoning the direct approach suggested by Central Limit Theorem and using the sampling distribution of $\ln LF$ instead. We also design tests to decide whether it is possible to reduce the number of the model's parameters without adversely affecting its predictive power. Finally, we modify the technique to cover the possibility of estimating only a subset of the model's parameters while ignoring the rest (known as **nuisance** parameters).

2. Autoregressive Model

In this model, the i^{th} observation (denoted X_i) is generated by a linear combination of the last m observations and an extra term ε_i, independent of these and Normally distributed with the mean of zero and standard deviation σ ($\mathcal{N}(0,\sigma)$ in our notation), *i.e.* by

$$X_i = \mu + \alpha_1 (X_{i-1} - \mu) + \alpha_2 (X_{i-2} - \mu) + \cdots + \alpha_m (X_{i-m} - \mu) + \varepsilon_i \tag{1}$$

where m is a small integer. When $m = 1$ or 2, the model is called *Markov* or *Yule* respectively; in all other cases, we refer to it as the *AR*(m) model. We study it in its *stationary* mode only, meaning that the process has equilibrated before we start observing it (this implies, among other things, that all X_i have the *same* Normal distribution). The value of the μ parameter is arbitrary, σ is positive, while the α values must meet (for the process to be stationary) three specific inequalities, introduced shortly (see [4]).

Note that, after standardizing the X_i by

$$Z_i := \frac{X_i - \mu}{\sigma} \tag{2}$$

(1) simplifies to

$$Z_i = \alpha_1 Z_{i-1} + \alpha_2 Z_{i-2} + \cdots + \alpha_m Z_{i-m} + \tilde{\varepsilon}_i \tag{3}$$

where $\tilde{\varepsilon}_i \in \mathcal{N}(0,1)$.

2.1. Variance and Serial Correlation

One can readily establish that the *joint* distribution of all the X_i ($i = 1, 2, \cdots, n$) random variables is multivariate Normal with the same mean of μ and a common variance which equals to σ^2 further multiplied by a function (5) of the α parameters. The (serial) correlation coefficient between X_i and X_{i+k} is the same regardless of the value of i (this follows from being stationary); we denote it ρ_k (note that $\rho_{-k} = \rho_k$).

To find the first m values of ρ, we need to solve the following set of linear equations (a routine exercise)

$$\rho_k = \sum_{j=1}^{m} \alpha_j \rho_{|k-j|} \quad \text{for } 1 \le k \le m \tag{4}$$

with the understanding that $\rho_0 = 1$, while the common variance is

$$V = \frac{1}{1 - \sum_{k=1}^{m} \alpha_k \rho_k} \tag{5}$$

The remaining ρ values (we need to go up to ρ_{n-1}) then follow from the following recurrence formula

$$\rho_k = \sum_{j=1}^{m} \alpha_j \rho_{k-j} \quad \text{when } k > m \tag{6}$$

while the $(k,\ell)^{\text{th}}$ element of the (n by n) variance-covariance (VC) matrix (which we denote by $\mathbb{V}$) of the X_i's equals to $\sigma^2 V \rho_{|k-\ell|}$.

Note that ρ_k, V and $\mathbb{V}$ remain the same for the Z_i sequence, while σ is then equal to 1.

These formulas are all well established in existing (and extensive) literature; it should suffice to quote [5] as our reference.

2.2. ML Estimation

To construct the corresponding multivariate probability density function (PDF), which becomes the likelihood function (denoted *LF*) when the X_i's are replaced by actual observations, we need the inverse of $\mathbb{V}$ and also its determinant. There is a general formula for elements of the inverse; getting the determinant is more difficult (note that symbolic computation on large matrices is practically impossible). Nevertheless, there is a way of bypassing this problem (see [6]) by finding the PDF of $Z_1, Z_2, \cdots, Z_m$, $\tilde{\varepsilon}_{m+1}, \tilde{\varepsilon}_{m+2}, \cdots, \tilde{\varepsilon}_n$ first (a simple exercise, since symbolic inversion and determinant computation of an m by m matrix becomes feasible) and then transforming it to get the PDF of the original set (equally simple: substitute (2) and divide by σ^n); this yields $\ln LF$ of the X sequence (while ignoring the constant $-\frac{n}{2}\ln(2\pi)$ term), namely

$$-\frac{\sum_{i,j=1}^{m} Z_i \left(\mathbb{V}^{-1}\right)_{i,j} Z_j + \sum_{i=m+1}^{n} \left(Z_i - \sum_{j=1}^{m} \alpha_j Z_{i-j}\right)^2}{2} - n \ln \sigma + \frac{\ln \det\left(\mathbb{V}^{-1}\right)}{2} \tag{7}$$

where $\mathbb{V}$ is now the VC matrix of only the first m terms of the Z_i sequence; the second sum is the standard Normal PDF of the $\tilde{\varepsilon}$ subsequence, after solving (3) for $\tilde{\varepsilon}_i$ and substituting.

As an example, we select (rather arbitrarily) $m = 3$, $\mu = 50$, $\sigma = 0.6$, $\alpha_1 = 2.7$, $\alpha_2 = -2.52$ and $\alpha_3 = 0.81$ and generate a random sequence of 300 consecutive and equilibrated observations using the corresponding *AR*(*3*) model of (1), and the following Mathematica code.

```
{m, n} = {3, 300}; X = RandomVariate[NormalDistribution[0, 0.6], n + 100];
Do[X[[i]] = 50 + 2.7 (X[[i - 1]] - 50) - 2.52 (X[[i - 2]] - 50)
   + 0.81 (X[[i - 3]] - 50) + X[[i]], {i, m + 1, n + 100}]
ListPlot[X = Drop[X, 100], AspectRatio -> .2, ImageSize -> 600,
   PlotStyle -> {Black, PointSize[.003]}]
```

The first line generates a sequence of $n+100$ *independent* values from the

$\mathcal{N}(0,0.6)$ distribution—the ε_i of (1); the second line (and its continuation) converts them to the X_i of the same formula, using $m=3$. The *extra* 100 observations were needed to allow the process to equilibrate (the first 100 X_i's are then deleted); the last line displays the resulting $AR(3)$ sequence in **Figure 1**.

Using the generated sequence as data from an $AR(3)$ model whose five parameters are unknown and to be estimated, we proceed to

- solve (4) for the first 3 (m in general) ρ's,
- compute V using (5),
- and the corresponding inverse of the $\mathbb{V}$ matrix (denoted A),
- standardize the observation in the manner of (2),
- build the $-\frac{2}{n}$ multiple of (7) while excluding its last term (the result is called L).

This is done (one line for each bullet) by the following continuation of the previous computer program:

```
ρ0 = 1; rho = Solve[Table[ ρj == Sum[αj ρAbs[i-j], {j, m}], {i, m}],
   Table[ρi, {i, m}]] [[1]];
V = (1 - Sum[ αi ρi, {i, m}])^-1 /. rho // Factor
```

$$\frac{1-\alpha_2-\alpha_1\alpha_3-\alpha_3^2}{(1-\alpha_1-\alpha_2-\alpha_3)(1+\alpha_1-\alpha_2+\alpha_3)(1+\alpha_2+\alpha_1\alpha_3-\alpha_3^2)}$$

```
A = Inverse[ Table[ρAbs[i-j] V, {i, m}, {j, m}]] /. rho // Factor;
Z = (X - μ)/σ];
L = 2 Log[σ] + Take[Z, m] . A . Take[Z, m] / n
   + Sum[(Z[[i]] - Sum[Z[[i - j]] αj, {j, m}])^2, {i, m + 1, n}] / n;
```

In addition to computing L we have also printed V, as its denominator provides a useful set of if-and-only-if conditions for the process to be stationary (all of its three factors must be positive).

Now, we need to either maximize $\ln LF$ (a difficult task), or make each of its derivatives (with respect to every parameter) equal to zero and solve the corresponding set of normal equations; the latter approach is easier, faster and more accurate; it involves the following steps:

- compute the five ($2+m$ in general) derivatives of L (the first two lines of the code below); multiply each by σ^2, thus making them *free* of σ (with the exception of the σ derivative which, when multiplied by σ^3 becomes *linear* in σ^2); denote the results $d\mu$, $d\sigma$ and $d\alpha$ (the last being a set of m expressions),

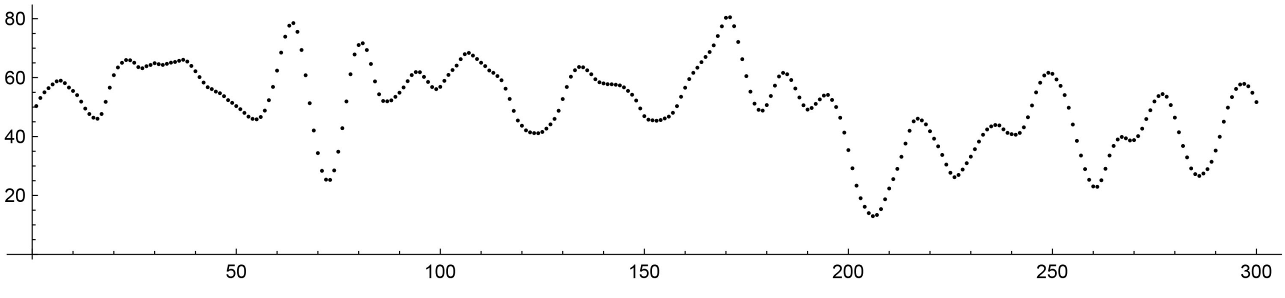


Figure 1. Randomly generated data based on $AR(3)$ model.

• similarly (line three), differentiate the extra $\frac{\ln\det\left(\mathbb{V}^{-1}\right)}{n}$ term of (7) with respect to each α parameter, denoting the answer dV (a collection of m functions of only the α parameters),

• set the initial value of μ to $\sum_{i=1}^{n} X_i/n$, of σ (arbitrarily) and of each α to 0, and start the following iteration (performed by Mathematica's routine *ite*):

• solve

$$d\alpha\big|_{\mu} = dV\big|_{\alpha} \tag{8}$$

for a *new*, updated set of α values (the second line of *ite*; its first line is only preparatory—note its continuation) where $d\alpha\big|_{\mu}$ implies that $d\alpha$ is evaluated using the *current* value of μ, while $dV\big|_{\alpha}$ is similarly evaluated using the *current* set of α values; the corresponding equations for the new α's are thus linear and easy to solve,

• solve

$$d\mu\big|_{\alpha} = 0 \tag{9}$$

to get a new value of μ, where the evaluation is now done using the updated α's (third line of *ite*); the equation is linear in μ and thus trivial to solve,

• finally, solve

$$d\sigma\big|_{\mu,\alpha} = 0 \tag{10}$$

for σ, where the evaluation is done using the new values of both μ and α (fourth line of *ite*; the final line just combines the updated values of μ, σ and α's into a single output); the equation is linear in σ^2 (its positive root yields the updated σ),

• repeat the iteration till the new values of the five parameters no longer change (a total of five iterations normally suffice to reach an adequate accuracy); this is done by the last line of the subsequent code, which also returns the resulting ML estimates.

To do this, we need to further extend the previous Mathematica code by:

```
dμ = σ^2 D[L, μ] // Expand; dσ = σ^3 D[L,σ] //Expand;
dα = Table[D[L, α_i], {i, m}] σ^2 //Expand;
dV = Table[ D[Log[Det[A]] σ^2/n // Factor, α_i], {i, m} ];
ite[pr_] := Module[ {μv = μ -> pr[[1]], σv = σ -> pr[[2]],
   αv = Thread[ (αi = Table[α_i, {i, m} ] ) -> Drop[pr, 2] ] },
αv = Solve[ (dα /. μv) == (dV /. αv /. σv), αi] [[1]];
μv[[2]] = μ /. Solve[(dμ == 0 /. αv), μ] [[1]];
σv[[2]] = Sqrt[- dσ/2 + σ^2 /. αv /. μv //Expand];
( {μ, σ, αi} // Flatten ) /. μv /. σv /. αv]
sol = Nest[ite, {Mean[X], 1, Table[0, m] } //Flatten, 5] ]
```

$\{50.094, 0.571, 2.650, -2.453, 0.793\}$

thus obtaining the ML estimates of μ, σ and the three α's (in that order).

An interesting observation: the last term of L (whose numerator is a sum of squares of differences between predicted and actual values of the Z sequence) always results in 1 when evaluated using the ML estimates returned by *sol*; this can serve as a verification of the program working correctly.

2.3. Asymptotic Distribution

The general version of central limit theorem (CLT) tells us that the sampling distribution of ML estimators is approximately Normal, with the asymptotic means equal to the parameters' true values. To get the distribution's VC matrix, we utilize the theory of Fisher information matrix and

- divide $-\ln LF$ by n, getting

$$\ln\sigma + \frac{1}{2}\frac{\sum_{i=1}^{n} Z_i^2}{n}\sum_{j=0}^{m}\alpha_j^2 + \sum_{k=1}^{m}\frac{\sum_{i=1}^{n-k} Z_i Z_{i+k}}{n}\sum_{j=0}^{m-k}\alpha_j\alpha_{j+k} \tag{11}$$

where $\alpha_0 = -1$; this follows from (7),

- find the matrix of second derivatives (with respect to each *pair* of parameters) of this expression,
- find the expected value (indicated by $\mathbb{E}$) of each element of this matrix, remembering that

$$\mathbb{E}(Z_i) = 0 \text{ and } \mathbb{E}(Z_i Z_{i+k}) = V\rho_k, \tag{12}$$

- take the corresponding $n \to \infty$ limit,
- invert the resulting matrix,
- and further divide by n.

Leaving out further details, this yields the following results: ML estimators of μ and σ are, asymptotically (*i.e.* when n is large) independent of each other and of the α estimators, with standard deviation given by

$$\frac{\sigma/\sqrt{n}}{1-\sum_{k=1}^{m}\alpha_k} \text{ and } \frac{\sigma}{\sqrt{2n}} \tag{13}$$

respectively, while the α part of the asymptotic VC matrix is

$$\frac{1-\alpha_1^2}{n}, \begin{bmatrix} 1-\alpha_2^2 & -\alpha_1(1-\alpha_2) \\ -\alpha_1(1-\alpha_2) & 1-\alpha_2^2 \end{bmatrix} / n \text{ and}$$
$$\begin{bmatrix} 1-\alpha_3^2 & -\alpha_1-\alpha_2\alpha_3 & -\alpha_2-\alpha_1\alpha_3 \\ -\alpha_1-\alpha_2\alpha_3 & 1+\alpha_1^2-\alpha_2^2-\alpha_3^2 & -\alpha_1-\alpha_2\alpha_3 \\ -\alpha_2-\alpha_1\alpha_3 & -\alpha_1-\alpha_2\alpha_3 & 1-\alpha_3^2 \end{bmatrix} / n \tag{14}$$

in the case of Markov ($m=1$), Yule ($m=2$) and *AR*(*3*) model, respectively. Note that each VC matrix is both symmetric and *slant*-symmetric (*i.e.* the same when flipped with respect to *either* diagonal). They have been found by the first line of the following Mathematica code (by specifying the value of m first).

When executed at the end of the last program (where m was already set to 3), the second line establishes the standard error (using (13) and (14), with parameter *estimates* replacing their true values) of each of the previously computed ML estimates; note that they are all within two standard errors of their true values.

```
α0 = -1; VC = Table[0, {i, m}, {i, m}]; Do[VC[[i, j]] = VC[[j, i]] =
   VC[[m + 1 - j, m + 1 - i]] = VC[[m + 1 - i, m + 1 - j]] =
   Sum[αi αl+j-i - αm-l αm+i-l-j, {l, 0, i - 1}], {i, m + 1}, {j, i, m + 1 - i}]
{σ/(1 - α1 - α2 - α3), σ/Sqrt[2],
   Sqrt[VC[[1, 1]]], Sqrt[VC[[2, 2]] ], Sqrt[VC[[3, 3]] ] }/Sqrt[n] /.
   Thread[Table[αi, {i, m}] -> sol[[3 ;; m + 2]] ] /. σ -> sol[[2]]
                    {3.309, 0.023, 0.035, 0.068, 0.035}
```

Using these results, we can then compute the standard error of the *expected value* of the next (yet to happen) observation, established based on (1), the last m observations, and our estimates of μ and the α's.

3. Confidence Regions

To construct a confidence region (CR) for true values of all $m+2$ parameters of an $AR(m)$ model, we could directly use their asymptotic Normal distribution of the previous section (see [7] [8] and [9]). Nevertheless, it is more accurate (in terms of establishing the correct level of confidence), and also notably easier, to utilize the theorem which states that

$$2\ln LF\big|_{MLE} - 2\ln LF\big|_{\text{true}} \tag{15}$$

has, to a good approximation, the chi-squared distribution with $m+2$ degrees of freedom (notation: χ^2_{m+2}). Here, the first term of (15) involves the *maximum* value of $\ln LF$ achieved by the ML estimates, while the second term assumes evaluating the same $\ln LF$ using the *true* values of the parameters. The proof of a similar statement can be found in [10]; the necessary modifications are quite simple and need not be elaborated on. The theorem's accuracy is demonstrated in **Figure 2**, which compares the empirical histogram of 10^5 randomly generated values of (15) to the PDF of χ^2_3, using a Markov model with $\mu=10$, $\sigma=1.7$, $\alpha_1=0.9$ and $n=300$.

Even though the agreement is not quite perfect (it quickly improves with increasing n), both the accuracy and simplicity of this approach greatly outweigh those of CLT (**Figure 6** demonstrates the error of the basic Normal approximation: it is far less accurate than what we see in **Figure 1**).

Utilizing this asymptotic distribution, we then make (15) equal to a selected critical value of χ^2_3; solution to this equation yields the CR boundaries. To show how it is done in the case of the above Markov model, we run the program of our original example while changing its first three lines to

```
{m, n} = {1, 300}; X = RandomVariate[NormalDistribution[0, 1.7], n + 100];
Do[X[[i]] = 10 + 0.9 (X[[i - 1]] - 10) + X[[i]], {i, m + 1, n + 100}]
ListPlot[X = Drop[X, 100], AspectRatio -> .2, PlotStyle -> Black,
  ImageSize->600, Joined -> True]
```

This returns a set of data displayed in **Figure 3** (the individual observations have been connected) and, after executing the program's next two segments, also the corresponding three ML estimates (saved in *sol*).

Computing L and *res* by the previous routines, and adding the following two lines of code (the first evaluating (15), the second doing the plotting)

```
aux = (- n L + Log[Det[A]] /. Thread[{μ, μ, α1} -> sol]) + n L - Log[Det[A]];
ContourPlot3D[ aux == InverseCDF[ChiDistribution[3], .95], {μ, 6, 13.5},
  {σ, 1.4, 2.2}, {α1, .8, 1}, ContourStyle -> Black, AxesLabel -> {μ, σ, α1}]
```

produces the desired CR, displayed in **Figure 4** (in Mathematica, it can be rotated and observed from any direction).

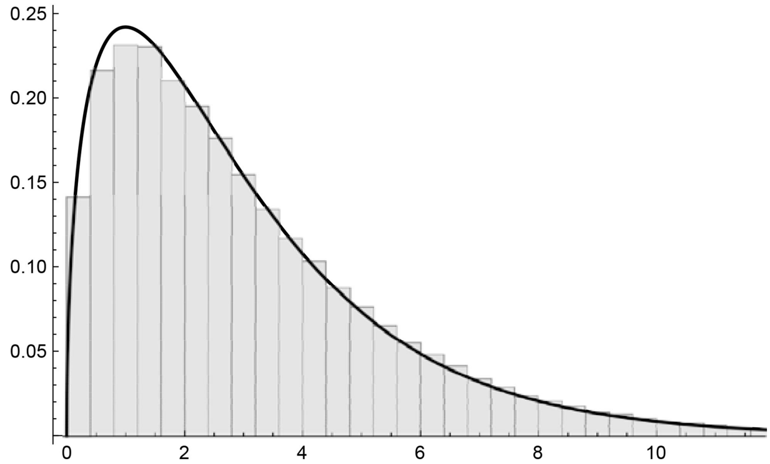


Figure 2. Empirical and theoretical distribution of (15).

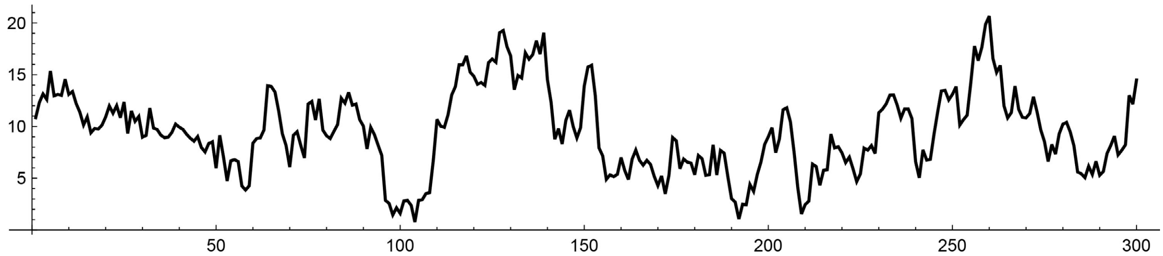


Figure 3. Randomly generated data from Markov model.

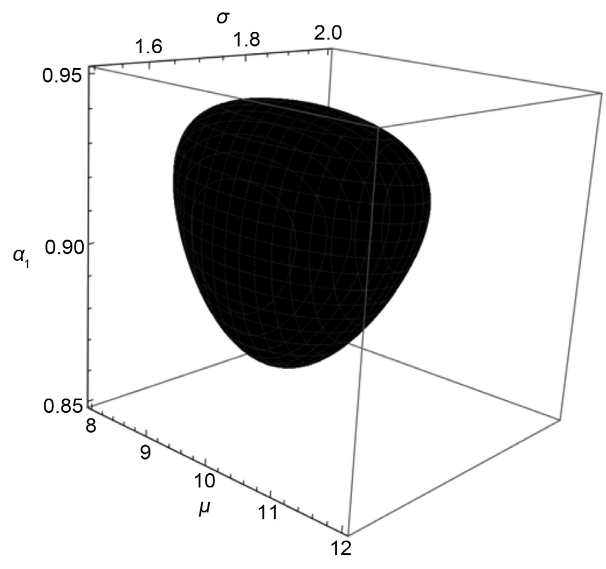


Figure 4. 95% confidence region for Markov-model parameters.

3.1. Nuisance Parameters

We have already shown that the three ML estimators of Markov-model parameters are asymptotically independent; this enables us to find a confidence interval (CI) for any *one* of these (we call it the *pivotal* parameter) by treating the other two as *nuisance* parameters (see [11]). The reference indicates that changing (15) to

$$2\ln LF\big|_{MLE} - 2\ln LF\big|_{\text{mixed}} \tag{16}$$

where "mixed" implies using ML estimates for the nuisance parameters and the true value of the pivotal one) makes (16) into a χ_1^2 type of random variable.

Using parameters of the Markov example of the last section, we display, in **Figure 5**, the empirical and theoretical PDF of (16), while considering μ and σ to be nuisance parameters and α_1 the pivotal one; the agreement is nearly perfect.

Let us contrast this result with similar comparison of the actual (empirical) distribution of the ML estimator of α_1 with its asymptotic (CLT-based) limit (Normal, with the mean of 0.9 and variance, based on (14), of $\left(1-0.9^2\right)/300$), which is displayed in **Figure 6**. This time the error of the approximation is clearly inacceptable; using it for constructing CI for the true value of α_1 is not recommended.

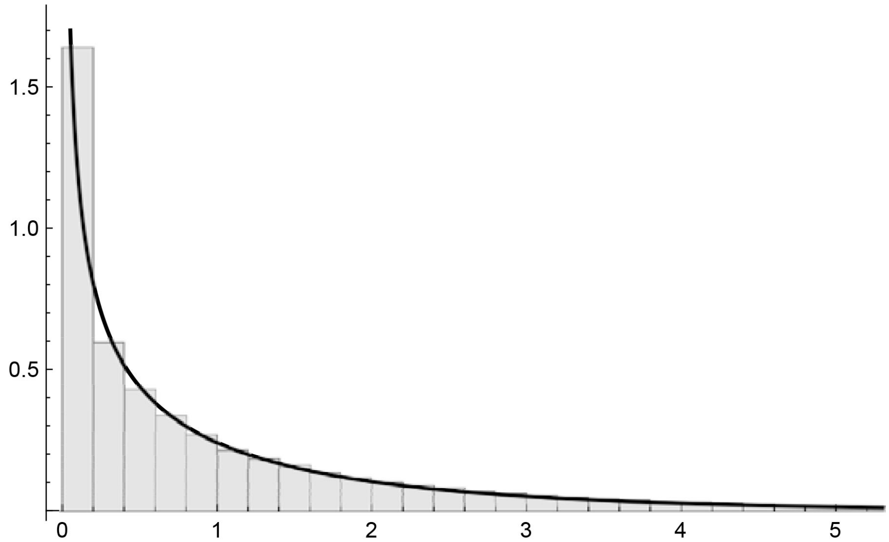


Figure 5. Empirical and asymptotic PDF of (16); nuisance parameters are μ and σ, m = 1.

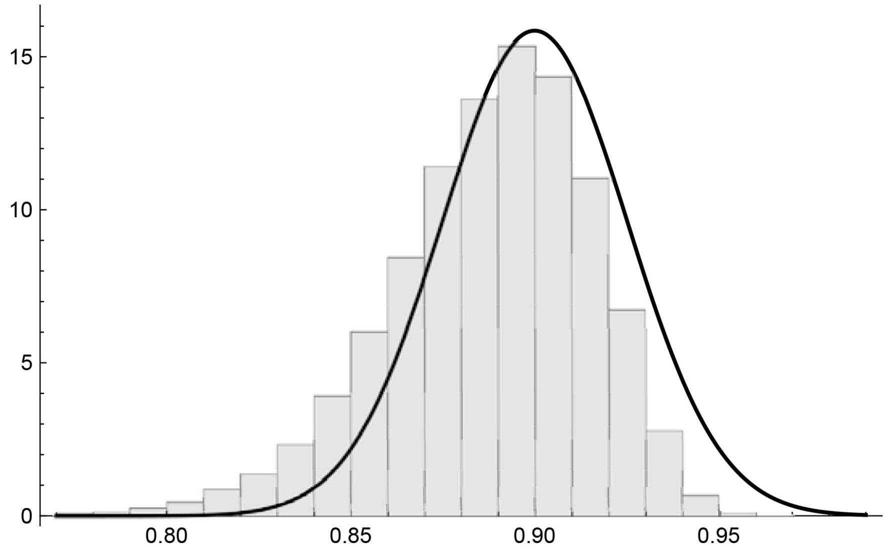


Figure 6. Empirical distribution of α_1, versus its Normal approximation.

To accurately compute boundaries of a CI for α_1, we must thus return to (16) and utilize its fast convergence to χ_1^2 distribution. First, we find ML estimates of *all* parameters (already done in previous section), then add the following two lines of code (the first line evaluating (16), the second one making it equal to the 80% critical value of χ_1^2 and solving this equation for α_1):

```
aux = (Log[Det[A]] - n L /. Thread[{μ, σ, α1} -> sol] )
  - (Log[Det[A]] - n L /. {μ -> sol[[1]], σ -> sol[[2]] } ) //Simplify;
NSolve[aux == InverseCDF[ChiSquareDistribution[1], 0.8], α1, Reals]
```

$$\{\{\alpha_1 -> 0.902792\}, \{\alpha_1 -> 0.956463\}\}$$

We then claim, with an 80% confidence, that the true value of α_1 lies between the resulting two roots.

Note that the theory allows us to designate *any* set of parameters as pivotal (the rest are then the nuisance parameters); when the two sets of estimators are asymptotically independent, the distribution of (16) is approximately χ_p^2, where p is the number of pivotals. As soon as there is a *non-zero* correlation *between* the two groups, the distribution of (16) becomes substantially more complicated and, in the case of $AR(m)$ models, practically impossible to deal with. The technique can thus be applied only to situations when all α parameters are either *all* pivotal or *all* nuisance, as demonstrated by our next example.

This time, we generate data from Yule model with $\mu = 10$, $\sigma = 2$, $\alpha_1 = 1.75$, $\alpha_2 = -0.9$, using $n = 100$, by the usual

```
{m, n} = {2, 100}; X = RandomVariate[NormalDistribution[0, 2], n + 100];
Do[X[[i]] = 10 + 1.75 (X[[i - 1]] - 10) - 0.9 (X[[i - 2]] - 10) + X[[i]],
  {i, m + 1, n + 100}]
ListPlot[X = Drop[X, 100], AspectRatio -> .2, ImageSize->600,
  PlotStyle -> {Black, PointSize[0.004] } ]
```

which produces the sequence of **Figure 7**.

We then run the common part of the program to find the four ML estimates, saving them under the name *res*. Executing the following two extra lines (the first line evaluating (16), the second one making it equal to the 95% critical value of χ_2^2, solving for α_1 and α_2, and displaying the resulting contour):

```
aux = (Log[Det[A]] - n L /. Thread[{μ, σ, α1, α2} -> sol])
  - (Log[Det[A]] - n L /. {μ -> sol[[1]], σ -> sol[[2]] } ) // Simplify;
ContourPlot[ aux == InverseCDF[ChiDistribution[2], .95], {α1, 1.71, 1.81},
  {α2, -0.96, -0.87}, ContourStyle -> Black, AxesLabel -> {α1, α2} ]
```

we get the 90% CR of **Figure 8** for the true values of α_1 and α_2, while considering μ and σ as nuisance parameters.

This is based on our previous assertion that the distribution of (16) is, in this case, approximately χ_2^2; to check the accuracy of this statement, we display the empirical version of this distribution (using the same Yule model and 10^5 sequences of 100 consecutive observations) together with the PDF of χ_2^2 in **Figure 9**; the agreement is again practically perfect.

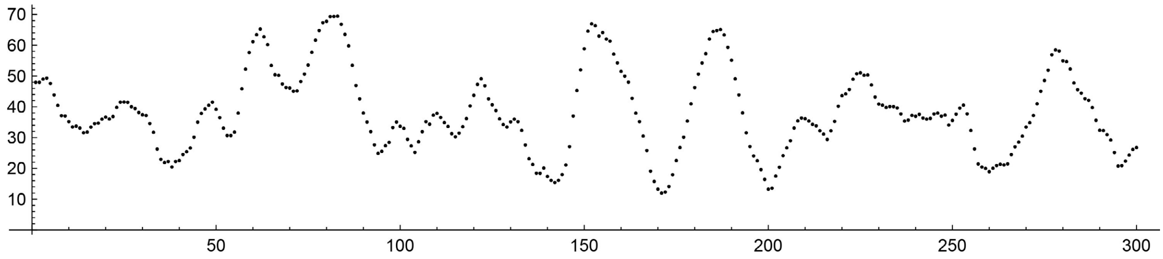


Figure 7. Randomly generated data from Yule model.

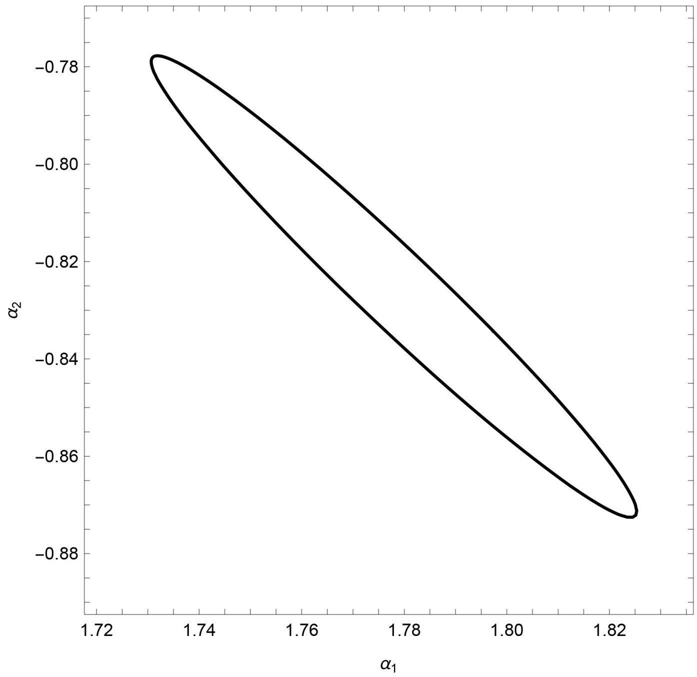


Figure 8. 90% confidence region for α_1 and α_2.

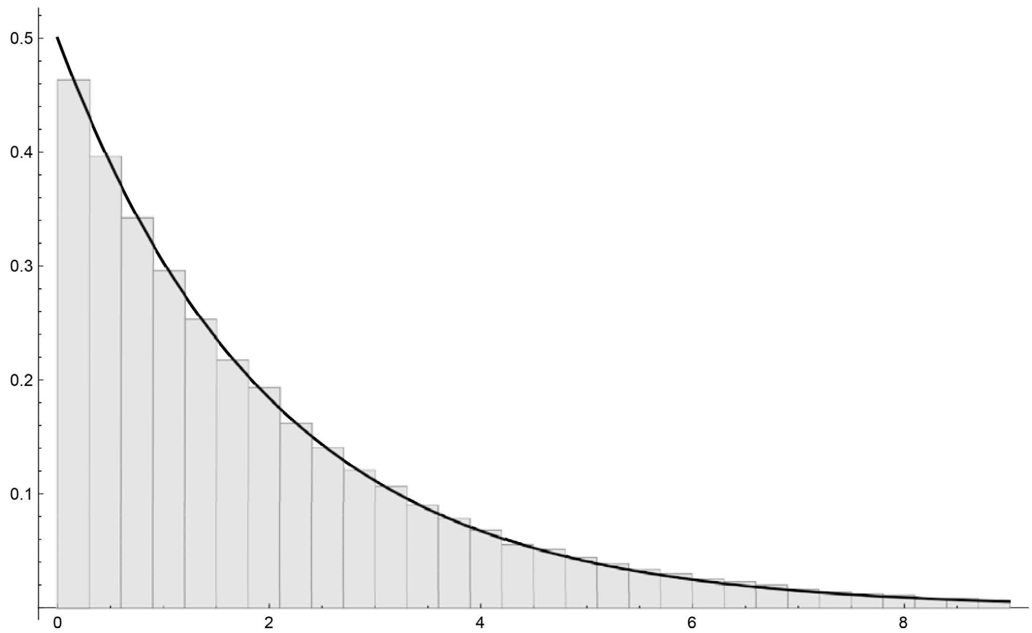


Figure 9. Empirical and theoretical PDF of (16); nuisance parameters are μ and σ, m = 2.

3.2. Establishing Model's Order

Until now, we always assumed that the value of m (the *order* of the model) is known; this may normally *not* be the case, so the question is: how do we test whether an $AR(m)$ model has the correct number of α parameters to be an adequate representation of a given set of consecutive observations? The obvious idea is to see if, by extending the model by an extra α_{m+1} parameter, the corresponding increase in the maximum value of the $\ln LF$ function is statistically significant. This can be easily tested by using an extension of the last theorem: when the data *does* follow an $AR(m)$ model, the following difference

$$2\ln LF\big|_{m} - 2\ln LF\big|_{m+1} \tag{17}$$

(where the subscript indicates the number of α parameters used to maximize $\ln LF$) has χ_1^2 distribution.

We verify this by yet another Monte-Carlo simulation of 10^5 sequences generated from the $AR(3)$ model of our first example, displaying empirical histogram of the resulting (17) values, together with the PDF of χ_1^2, in **Figure 10**. The agreement indicates that the latter distribution is again an excellent approximation of the former.

Returning to the original example; we have already computed ML estimates of the five parameters, which are easily converted into the corresponding value of $\ln FL$. We then need to use the same data and run the same program with $m = 4$, similarly converting the resulting *six* estimates to a new (always higher) value of $\ln LF$. Evaluating (17) and substituting the result into the CDF of χ_1^2 (we let the reader come up with the corresponding code) yields 0.111; this, being less than 95%, tells us that the data is adequately described by the original $AR(3)$ model.

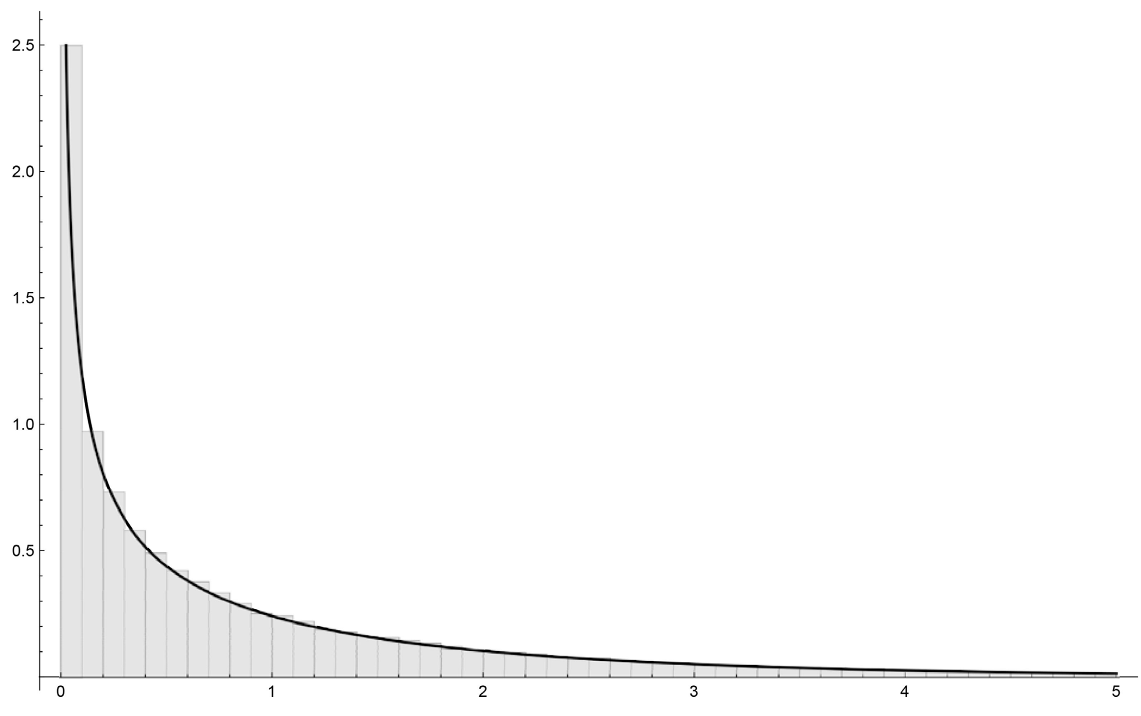


Figure 10. Empirical and asymptotic PDF of (17); $m = 3$.

On the other hand, trying to fit a Yule model ($m = 2$) to the same data yields substantially higher difference between $\ln LF|_2$ and $\ln LF|_3$, resulting (using the same example) in a CDF value indistinguishable from 1 (the corresponding P-value is of the order of 10^{-61}), indicating that no Yule model can properly describe this data.

We mention in passing that, to test the same $\alpha_3 = 0$ hypothesis, we could resort to a less sophisticated approach of using the asymptotic distribution of the ML estimator of α_3 (Normal, with the mean of 0 and variance equal to $\frac{1}{n}$, under the null hypothesis), getting a similarly small P-value of 10^{-40}; this test is not only less accurate (the Normal approximation has a large error, as we have seen) but also less sensitive than the one based on (17).

4. Conclusions

In this article, we have provided a variety of techniques for estimating parameters of an $AR(m)$ model, including construction of confidence intervals/regions and testing various hypotheses regarding parameter values and the model's order. We have used Monte Carlo simulation extensively to find empirical distributions of various sample statistics and explore the accuracy of each proposed approximation. This led us to conclude that the traditional use of CLT and of the corresponding Normal distribution is to be discouraged due to the resulting large errors. On the other hand, utilizing differences between various $\ln LF$ functions and the corresponding χ^2 distribution yielded very accurate results even with relatively small sets of past observations.

There are several directions to pursue in terms of potential future research, e.g.: investigating how flexible $AR(m)$ models are to describe, to a sufficient accuracy, *general* stationary stochastic processes (such as moving averages) that do not necessarily originate from (1); after all, real data do not exactly follow any mathematical model—these function only as useful approximations. Another possible extension of the current research would be to similarly deal with ARMA models; here, one would need to find some practical way of inverting (and finding determinants) of large symbolic matrices, essential for computing the corresponding $\ln LF$ function.

Conflicts of Interest

The author declares no conflicts of interest regarding the publication of this paper.

References

[1] Kendall, M.G. and Stuart. A. (1968) The Advanced Theory of Statistics. Vol. 3, 2nd Edition, Hefner Publishing Company, Los Angeles.

[2] Armour, B. and Morgera, S.D. (1991) An Exact Forward-Backward Maximum Likelihood Parameter Estimation Method. *IEEE Transactions on Signal Processing*, **39**, 1985-1993. https://doi.org/10.1109/78.134431

[3] Liu, X. and Peng, L. (2019) Asymptotic Theory and Unified Confidence Region for an Autoregressive Model. *Journal of Time Series Analysis*, **40**, 43-65. https://doi.org/10.1111/jtsa.12418

[4] Vrbik, J. and Vrbik, P. (2013) Informal Introduction to Stochastic Processes with Maple. Springer, New York. https://doi.org/10.1007/978-1-4614-4057-4

[5] Vrbik, J. (2017) General Formulas for Serial Correlation, Variance and Likelihood Function Relating to AR(k) Models. *Advances and Applications in Statistics*, **50**, 411-421. https://doi.org/10.17654/AS050050411

[6] Brockwell, P.J. and Davis, R.A. (2002) Introduction to Time Series and Forecasting, 2nd Edition, Springer, New York. https://doi.org/10.1007/b97391

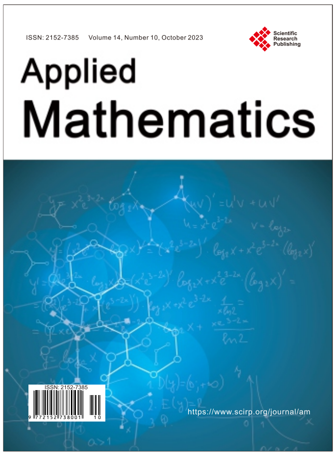
[7] Phillips, P.C.B. (2014) On Confidence Intervals for Autoregressive Roots and Predictive Regression. *Econometrica*, **82**, 1177-1195. https://doi.org/10.3982/ECTA11094

[8] Mikusheva, A. (2007) Uniform Inference in Autoregressive Models. *Econometrica*, **75**, 1411-1452. https://doi.org/10.1111/j.1468-0262.2007.00798.x

[9] Elliott, G. and Stock. J.H. (2001) Confidence Intervals for Autoregressive Coefficients near One. *Journal of Econometrics*, **103**, 155-181. https://doi.org/10.1016/S0304-4076(01)00042-2

[10] Vrbik, J. (2022) High-Accuracy Confidence Regions for Distribution Parameters. *Applied Mathematics*, **13**, 488-501. https://doi.org/10.4236/am.2022.136031

[11] Vrbik, J. (2022) Confidence Regions with Nuisance Parameters. *Open Journal of Statistics*, **12**, 658-675. https://doi.org/10.4236/ojs.2022.125039



Call for Papers

Applied Mathematics (AM)

ISSN Print: 2152-7385 ISSN Online: 2152-7393
https://www.scirp.org/journal/am

Applied Mathematics **(AM)** is an international journal dedicated to the latest advancements in applied mathematics. The goal of this journal is to provide a platform for scientists and academicians all over the world to promote, share, and discuss various new issues and developments in different areas of applied mathematics.

Subject Coverage

All manuscripts must be prepared in English, and are subject to a rigorous and fair peer-review process. Accepted papers will immediately appear online followed by printed hard copy. The journal publishes original papers including but not limited to the following fields:

- Applied Probability
- Applied Statistics
- Approximation Theory
- Chaos Theory
- Combinatorics
- Complexity Theory
- Computability Theory
- Computational Methods in Mechanics and Physics
- Continuum Mechanics
- Control Theory
- Cryptography
- Discrete Geometry
- Dynamical Systems
- Elastodynamics
- Evolutionary Computation
- Financial Mathematics
- Fuzzy Logic
- Game Theory
- Graph Theory
- Information Theory
- Inverse Problems
- Linear Programming
- Mathematical Biology
- Mathematical Chemistry
- Mathematical Economics
- Mathematical Physics
- Mathematical Psychology
- Mathematical Sociology
- Matrix Computations
- Neural Networks
- Nonlinear Processes in Physics
- Numerical Analysis
- Operations Research
- Optimal Control
- Optimization
- Ordinary Differential Equations
- Partial Differential Equations
- Probability Theory
- Statistical Finance
- Stochastic Processes
- Theoretical Statistics

We are also interested in: 1) Short Reports—2-5 page papers where an author can either present an idea with theoretical background but has not yet completed the research needed for a complete paper or preliminary data; 2) Book Reviews—Comments and critiques.

Notes for Intending Authors

Submitted papers should not have been previously published nor be currently under consideration for publication elsewhere. Paper submission will be handled electronically through the website. All papers are refereed through a peer review process. For more details about the submissions, please access the website.

Website and E-mail

https://www.scirp.org/journal/am **E-mail: am@scirp.org**