

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ  
імені Ігоря СІКОРСЬКОГО»  
НАВЧАЛЬНО-НАУКОВИЙ ФІЗИКО-ТЕХНІЧНИЙ  
ІНСТИТУТ

Кафедра математичного моделювання та аналізу даних

«До захисту допущено»

В.о. завідувача кафедри

\_\_\_\_\_ Іван ТЕРЕЩЕНКО

«\_\_» \_\_\_\_\_ 2023 р.

## Дипломна робота

на здобуття ступеня бакалавра

зі спеціальності: 113 Прикладна математика  
на тему: «Оцінювання параметрів частково спостережуваного  
ланцюга Маркова на двійкових послідовностях»

Виконав: студент 4 курсу, групи ФІ-91  
Цибульник Антон Владиславович

Керівник: ст. викл. Наказной П. О.

Консультант: к.ф.-м.н., доцент Ніщенко І. І.

Рецензент: (згодом)

Засвідчую, що у цій дипломній  
роботі немає запозичень з праць  
інших авторів без відповідних  
посилань.

Студент \_\_\_\_\_

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ  
імені Ігоря СІКОРСЬКОГО»  
НАВЧАЛЬНО-НАУКОВИЙ ФІЗИКО-ТЕХНІЧНИЙ  
ІНСТИТУТ

Кафедра математичного моделювання та аналізу даних

Рівень вищої освіти — перший (бакалаврський)  
Спеціальність (освітня програма) — 113 Прикладна математика,  
ОПП «Математичні методи моделювання, розпізнавання образів та  
комп'ютерного зору»

ЗАТВЕРДЖУЮ

В.о. завідувача кафедри

\_\_\_\_\_ Іван ТЕРЕЩЕНКО

«\_\_» \_\_\_\_\_ 2023 р.

**ЗАВДАННЯ**  
**на дипломну роботу**

Студент: Цибульник Антон Владиславович

1. Тема роботи: *«Оцінювання параметрів частково спостережуваного ланцюга Маркова на двійкових послідовностях»*,  
керівник: ст. викл. Наказной П. О.,  
затверджені наказом по університету №\_\_ від «\_\_» \_\_\_\_\_ 2023 р.
2. Термін подання студентом роботи: «\_\_» \_\_\_\_\_ 2023 р.
3. Вихідні дані до роботи: *опубліковані джерела за тематикою дослідження.*
4. Зміст роботи: *спостерігаючи часткову інформацію про динаміку бінарних послідовностей, за допомогою математичного апарату прихованих марковських моделей та із використанням методів математичної статистики досліджено деякі характеристики заданої моделі, зокрема: оцінено керуючий параметр динаміки системи; оцінено керуючий параметр динаміки системи у випадку додаткового зашумлення спостережуваних даних.*
5. Перелік ілюстративного матеріалу (із зазначенням плакатів, презентацій тощо): *презентація доповіді.*

6. Дата видачі завдання: 9 жовтня 2022 р.

### Календарний план

№ з/п	Назва етапів виконання дипломної роботи	Термін виконання	Примітка
1	Узгодження теми роботи із науковим керівником	09-14 жовтня 2022 р.	Виконано
2	Огляд опублікованих джерел за тематикою дослідження	15-30 жовтня 2022 р.	Виконано
3	Моделювання еволюції прихованої марковської моделі	Листопад 2022 р.	Виконано
4	Побудова оцінки для керуючого параметра системи	Грудень 2022 р.	Виконано
5	Побудова оцінки для керуючого параметра системи у випадку додаткового зашумлення спостережуваних даних	Січень 2023 р.	Виконано
6	Дослідження інших характеристик побудованої прихованої марковської моделі	Лютий 2023 р.	Виконано
7	Побудова статистичних оцінок невідомих параметрів моделі	Березень 2023 р.	Виконано
8	Програмна реалізація алгоритмів	Квітень-травень 2023 р.	Виконано

Студент

\_\_\_\_\_ Цибульник А. В.

Керівник

\_\_\_\_\_ Наказной П. О.

## РЕФЕРАТ

Кваліфікаційна робота містить: ??? стор., ??? рисунки, ??? таблиць, ??? джерел.

Об'єктом дослідження є ланцюг Маркова зі значеннями в множині двійкових послідовностей фіксованої довжини. Динаміка ланцюга задається як випадкове блукання вершинами одиничного куба, розмірність якого збігається з довжиною двійкової послідовності.

Стани заданого ланцюга є неспостережуваними (прихованими). Спостережуваними величинами в кожен момент часу є набір значень певного функціонала від фіксованих підмножин двійкової послідовності, яка описує поточний стан прихованого ланцюга.

Метою дослідження є побудова оцінок невідомих параметрів заданої марковської моделі за допомогою математичного апарату прихованих марковських моделей та із використанням методів математичної статистики.

Результати чисельного експерименту продемонстрували ефективність використаних методів, зокрема збіжність побудованих оцінок до істинних значень параметрів при збільшенні кількості спостережень.

ЛАНЦЮГ МАРКОВА, ПРИХОВАНА МАРКОВСЬКА МОДЕЛЬ,  
АЛГОРИТМ БАУМА-ВЕЛША, АЛГОРИТМ ВІТЕРБІ

## ABSTRACT

Qualification work contains: ??? pages, ??? figures, ??? tables, ??? sources.

The object of the study is a Markov chain with values in a set of binary sequences of fixed length. Chain dynamics is given as a random walk on the vertices of a unit cube whose dimension coincides with the length of the binary sequence.

The states of a given chain are unobservable (hidden). The observed values at each time point are a set of values of a certain functional from fixed subsets of the binary sequence that describes the current state of the hidden chain.

The purpose of the study is to construct estimates of unknown parameters of a given Markov model using the mathematical apparatus of hidden Markov models and using methods of mathematical statistics.

The results of the numerical experiment demonstrated the effectiveness of the methods used, in particular the convergence of constructed estimates to the true values of parameters with an increase in the number of observations.

MARKOV CHAIN, HIDDEN MARKOV MODEL, BAUM-WELCH ALGORITHM, VITERBI ALGORITHM

## ЗМІСТ

Перелік умовних позначень, скорочень і термінів .....	8
Вступ.....	9
1 Методи дослідження прихованих марковських моделей .....	11
1.1 Основні поняття і властивості ланцюгів Маркова .....	11
1.2 Поняття прихованої марковської моделі .....	12
1.3 Ітераційний алгоритм Баума-Велша.....	13
1.4 Алгоритм Вітербі .....	16
1.5 Властивості точкових оцінок .....	17
Висновки до розділу 1.....	19
2 Оцінка параметрів моделі.....	20
2.1 Моделювання об'єкта дослідження .....	20
2.2 Задача навчання .....	24
2.2.1 Ітераційний алгоритм Баума-Велша .....	24
2.2.2 Побудова оцінки методами математичної статистики.....	28
2.3 Задача декодування .....	29
2.4 Задача локалізації .....	29
2.4.1 Оцінка потужності шуканої множини .....	30
2.4.2 Оцінка шуканої множини.....	32
2.5 Задача навчання за спотвореними спостереженнями .....	34
Висновки до розділу 2.....	41
3 Проведення чисельного експерименту.....	42
3.1 Моделювання об'єкта дослідження .....	42
3.2 Задача навчання .....	44
3.2.1 Ітераційний алгоритм Баума-Велша .....	44
3.2.2 Побудова оцінки методами математичної статистики.....	45
3.2.3 Висновки щодо точності оцінювання.....	46
3.3 Задача декодування .....	47
3.4 Задача локалізації .....	49

3.5	Задача навчання за спотвореними спостереженнями .....	7 50
3.5.1	Ітераційний алгоритм Баума-Велша .....	51
3.5.2	Висновки щодо точності оцінювання.....	52
	Висновки до розділу 3.....	55
	Висновки .....	56
	Перелік посилань .....	57
	Додаток А Графічний інтерфейс користувача.....	58
	Додаток Б Тексти програм.....	60
Б.1	Обчислення коефіцієнтів прямого та зворотного ходу .....	60
Б.2	Алгоритм Баума-Велша.....	62
Б.3	Алгоритм Вітербі .....	64
Б.4	Алгоритм розв'язку задачі локалізації.....	65

## ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

$\mathbb{1}$  — індикаторна функція;

$\forall$  — квантор загальності: будь-який або для всіх;

$C_n^k$  — біноміальний коефіцієнт;

ПММ — прихована марковська модель.



## ВСТУП

**Актуальність дослідження.** Марковські моделі мають широкий та ефективний арсенал інструментів для аналізу динаміки систем, поведінка яких у кожен наступний момент часу зумовлюється лише поточним станом системи та не залежить від характеру еволюції у попередні моменти часу.

Водночас у випадку, коли безпосереднє спостереження еволюції ланцюга Маркова є неможливим чи обмеженим, застосовують моделі прихованих ланцюгів Маркова (ПММ). У такому випадку аналіз поведінки процесу відбувається за деякою опосередкованою інформацією про «приховані», справжні стани ланцюга.

Наприклад, в біоінформатиці [1, глава 9] апарат ланцюгів Маркова застосовують при дослідженні еволюції молекул ДНК протягом певного часу, вважаючи при цьому за стан системи зв'язану послідовність так званих нуклеотидів, які формуються над алфавітом чотирьох азотистих основ  $\{T, C, A, G\}$ .

Існування статистичних залежностей в чергуванні фонем чи слів в природних мовах зумовлює ефективність використання прихованих марковських моделей до таких завдань, як створення голосових команд, служб транскрипції та голосових помічників [2].

Не винятком стають і задачі розпізнавання мови жестів [3]: наприклад, представляючи жести як послідовності прихованих станів, ПММ можуть розпізнавати динаміку та варіації рухів рук.

Відтак, враховуючи актуальність вивчення еволюції систем, стани яких є послідовностями чи наборами символів певної довжини, у роботі розглядається ланцюг Маркова на множині двійкових послідовностей, динаміка якого відстежується за зміною в часі набору функціоналів від його станів.

**Метою дослідження** є побудова оцінок невідомих параметрів частково спостережуваного ланцюга Маркова на бінарних послідовностях.

Для досягнення мети необхідно розв'язати **задачу дослідження**, яка полягає у вирішенні таких завдань:

- 1) провести огляд опублікованих джерел за тематикою дослідження;
- 2) перевірити, чи задана модель відповідає необхідним умовам для використання апарату прихованих марковських моделей;
- 3) побудувати оцінки згідно з обраними методами;
- 4) експериментально перевірити ефективність отриманих оцінок.

*Об'єктом дослідження* є процеси, які описуються моделями частково спостережуваних ланцюгів Маркова.

*Предметом дослідження* є оцінки параметрів частково спостережуваного ланцюга Маркова на бінарних послідовностях.

При розв'язанні поставлених завдань використовувались такі *методи дослідження*: методи лінійної алгебри, теорії імовірностей, математичної статистики, методи комп'ютерного та статистичного моделювання, методи та алгоритми дослідження прихованих марковських моделей.

**Наукова новизна** отриманих результатів полягає у виведенні формул оцінювання невідомих параметрів системи за опосередкованими даними про еволюцію бінарних послідовностей.

**Практичне значення** результатів полягає у можливості їхнього застосування при аналогічних до умов дослідження припущеннях в аналізі процесів динаміки послідовностей, елементи яких мають дихотомічну структуру: наявність чи відсутність сигналів зв'язку, рух курсів акцій вниз чи вгору, коректність чи хибність символів комп'ютерного розпізнавача тощо.

**Апробація результатів та публікації.** Частина роботи була представлена на XXI Всеукраїнській науково-практичній конференції студентів, аспірантів та молодих вчених «Теоретичні і прикладні проблеми фізики, математики та інформатики» (11-12 травня 2023 р., м. Київ).

# 1 МЕТОДИ ДОСЛІДЖЕННЯ ПРИХОВАНИХ МАРКОВСЬКИХ МОДЕЛЕЙ

У цьому розділі будуть окреслені основні поняття та методи, які використовуватимуться у подальших викладках при розв’язуванні поставлених в межах дипломної роботи завдань.

## 1.1 Основні поняття і властивості ланцюгів Маркова

Нехай  $\{X^t\}_{t \geq 1}$  — послідовність випадкових величин зі значеннями в скінченній або зліченній множині  $E = \{e_1, e_2, \dots\}$ .

**Означення 1.1.** Послідовність  $\{X^t\}_{t \geq 1}$  утворює ланцюг Маркова, якщо:

$$\forall t \geq 2 \quad \forall i^1, i^2, \dots, i^{t+1} \in E : \\ P(X^{t+1} = i^{t+1} | X^t = i^t, \dots, X^1 = i^1) = P(X^{t+1} = i^{t+1} | X^t = i^t)$$

Цю умову називають марковською властивістю.

Множина  $E$  називається множиною станів ланцюга, а випадкова величина  $X^t$  трактується як стан системи в момент часу  $t$ . Надалі у ході дослідження по замовчуванню розглядатимуться так звані однорідні ланцюги Маркова, для яких ймовірності переходу  $P(X^{t+1} = j | X^t = i)$  з одного стану  $i \in E$  в інший  $j \in E$  не залежать від  $t$ .

Ймовірність  $p_{ij} = P(X^{t+1} = j | X^t = i)$  називається перехідною ймовірністю однорідного ланцюга Маркова. Матриця  $A$ , складена із цих ймовірностей, називається матрицею перехідних ймовірностей:

$$A = (p_{ij})_{i,j \in E} = (P(X^{t+1} = j | X^t = i))_{i,j \in E}$$

Ця матриця є стохастичною, тобто

$$\forall i, j \in E : p_{ij} \geq 0 \quad \text{та} \quad \forall i \in E : \sum_{j \in E} p_{ij} = 1$$

Окрім матриці  $A$ , для ланцюга Маркова слід задати вектор початкового розподілу ймовірностей

$$\pi = \left( \pi_i \right)_{i \in E} = \left( P(X^1 = i) \right)_{i \in E}$$

Надалі позначатимемо ланцюг Маркова з матрицею перехідних ймовірностей  $A$  та початковим розподілом  $\pi$  як  $\lambda = (\pi, A)$ .

Скінченновимірні розподіли [4] ланцюга Маркова  $\lambda = (\pi, A)$  повністю визначаються матрицею перехідних ймовірностей  $A$  та вектором початкового розподілу ймовірностей  $\pi$ . А саме:

$$\begin{aligned} \forall t \geq 2 \quad \forall i^1, i^2, \dots, i^t \in E : \\ P(X^1 = i^1, X^2 = i^2, \dots, X^{t-1} = i^{t-1}, X^t = i^t) = \pi_{i^1} \cdot p_{i^1 i^2} \cdot \dots \cdot p_{i^{t-1} i^t} \end{aligned} \quad (1.1)$$

## 1.2 Поняття прихованої марковської моделі

Розглянемо послідовність випадкових величин  $\{X^t\}_{t \geq 1}$  на скінченній або зліченній множині станів  $E = \{e_1, e_2, \dots\}$  та нехай  $\{Y^t\}_{t \geq 1}$  — послідовність випадкових величин на скінченній або зліченній множині  $F = \{f_1, f_2, \dots\}$ .

**Означення 1.2.** Пара  $\{(X^t, Y^t)\}_{t \geq 1}$ , задана на декартовому добутку  $E \times F$ , є прихованою марковською моделлю за виконання таких умов:

- 1) послідовність  $\{X^t\}_{t \geq 1}$  утворює ланцюг Маркова з початковим розподілом  $\pi$  та матрицею перехідних ймовірностей  $A$ ;
- 2) послідовність  $\{(X^t, Y^t)\}_{t \geq 1}$  є ланцюгом Маркова;
- 3) випадкові величини  $Y^1, Y^2, \dots, Y^t$  є умовно незалежними при

заданому наборі величин  $X^1, X^2, \dots, X^t$  :

$$\begin{aligned} \forall t \geq 2 \quad \forall j^1, j^2, \dots, j^t \in F \quad \forall i^1, i^2, \dots, i^t \in E : \\ P(Y^1 = j^1, \dots, Y^t = j^t \mid X^1 = i^1, \dots, X^t = i^t) = \\ = \prod_{k=1}^t P(Y^k = j^k \mid X^1 = i^1, \dots, X^t = i^t) \end{aligned}$$

4) умовний розподіл випадкової величини  $Y^k$  в момент часу  $k$  при заданих  $X^1, X^2, \dots, X^t$  залежить лише від  $X^k$  :

$$\begin{aligned} \forall t \geq 2 \quad \forall k = \overline{1, t} \quad \forall j^k \in F \quad \forall i^1, i^2, \dots, i^t \in E : \\ P(Y^k = j^k \mid X^1 = i^1, \dots, X^t = i^t) = P(Y^k = j^k \mid X^k = i^k) \end{aligned}$$

У парі  $\{(X^t, Y^t)\}_{t \geq 1}$  послідовність  $\{X^t\}_{t \geq 1}$  називають «прихованою», а послідовність  $\{Y^t\}_{t \geq 1}$  — «спостережуваною».

Окрім матриці  $A$  та вектора  $\pi$ , які задають ланцюг Маркова  $\{X^t\}_{t \geq 1}$ , прихована марковська модель визначається матрицею умовних імовірностей спостережень  $j \in F$  при заданих прихованих станах  $i \in E$  :

$$B = \left( B_{ij} \right)_{i, j \in E \times F} = \left( P(Y^t = j \mid X^t = i) \right)_{i, j \in E \times F}$$

Позначатимемо приховану марковську модель з початковим розподілом  $\pi$ , матрицею перехідних імовірностей  $A$  та матрицею умовних імовірностей спостережень при заданих прихованих станах  $B$  таким чином:  $\lambda = (\pi, A, B)$ .

### 1.3 Ітераційний алгоритм Баума-Велша

Нехай протягом деякого часу  $t = \overline{1, T}$  спостерігається послідовність випадкових величин

$$(Y^1 = y^1, \dots, Y^T = y^T) \iff Y = y$$

деякої прихованої марковської моделі  $\lambda = (\pi, A, B)$ , параметри  $\pi, A, B$  якої є невідомими. Постає питання: як за набором наявних даних віднайти оптимальні параметри моделі?

Скористаємося методом максимальної правдоподібності, шукаючи оцінку  $\hat{\lambda} = (\hat{\pi}, \hat{A}, \hat{B})$  шляхом максимізації ймовірності вигляду:

$$\hat{\lambda} = \underset{\lambda}{\operatorname{argmax}} P(Y = y | \lambda)$$

Інакше кажучи, шукатимемо такі параметри моделі, які найкраще пояснюють отримані спостереження  $y$ .

Ймовірність  $P(Y = y | \lambda)$  називається функцією правдоподібності. Задача максимізації цієї функції є складною чи неможливою в цілому через громіздкість отриманого виразу: зважаючи на вигляд (1.1) скінченновимірних розподілів ланцюга Маркова та враховуючи умови 3) й 4) означення прихованої марковської моделі (Озн. 1.2), функція правдоподібності набуває вигляду:

$$P(Y = y | \lambda) = \sum_{x \in E^T} P(X = x, Y = y | \lambda) = \sum_{x \in E^T} \pi_{x^1} \cdot \prod_{t=1}^{T-1} A_{x^t x^{t+1}} \cdot \prod_{t=1}^T B_{x^t y^t} \quad (1.2)$$

Безпосередня максимізація за цією формулою вимагає охоплення близько  $T \cdot |E|^T$  множників, де  $T$ , як правило, є великим. Однак, для прихованих марковських моделей можна застосувати інший підхід: модифікацію ЕМ-алгоритму [1, розділ 4] для дослідження прихованих ланцюгів Маркова — ітераційний алгоритм Баума-Велша [1, розділ 15].

Задавши деяке наближення невідомої моделі  $\lambda^{(0)} = (\pi^{(0)}, A^{(0)}, B^{(0)})$ , покладемо для наступної ітерації  $n + 1$

$$\lambda^{(n+1)} = \underset{\lambda}{\operatorname{argmax}} Q(\lambda^{(n)}, \lambda),$$

де

$$Q(\lambda^{(n)}, \lambda) = \sum_{x \in E^T} L_{\lambda^{(n)}} \ln L_{\lambda} \quad (1.3)$$

є так званою функцією квазі-log правдоподібності, а вираз

$$L_\lambda \equiv P(X = x, Y = y | \lambda)$$

називається функцією повної правдоподібності.

Доведено [1, розділ 4], що така ітераційна процедура є збіжною і приводить до точки локального максимуму логарифма функції правдоподібності (1.2). А оскільки точки екстремумів довільної функції та її логарифму збігаються, ця процедура розв'язує поставлену задачу.

Особливістю алгоритму навчання Баума-Велша є використання так званих змінних прямого (1.4) та зворотного (1.5) ходу, за допомогою яких обчислення функції правдоподібності вимагає лише  $T \cdot |E|^2$  добутків. Вказані коефіцієнти визначаються наступним чином:

$$\forall x \in E :$$

$$\alpha_t(x) = P(Y^1 = y^1, \dots, Y^t = y^t, X^t = x | \lambda^{(n)}) \quad (1.4)$$

$$\beta_t(x) = P(Y^{t+1} = y^{t+1}, \dots, Y^T = y^T | X^t = x, \lambda^{(n)}) \quad (1.5)$$

Перевага цих коефіцієнтів полягає у тому, що їх можна обчислити рекурентно [5, розділ 5] згідно з наведеними нижче співвідношеннями для змінних прямого ходу

$$\begin{aligned} t = 1 & \quad \forall x \in E : \alpha_1(x) = \pi_x B_{x,y^1} \\ t = \overline{2, T} & \quad \forall x \in E : \alpha_t(x) = \sum_{x' \in E} \alpha_{t-1}(x') A_{xx'} B_{xy^t} \end{aligned}$$

та змінних зворотного ходу

$$\begin{aligned} t = T & \quad \forall x \in E : \beta_T(x) = 1 \\ t = \overline{T-1, 1} & \quad \forall x \in E : \beta_t(x) = \sum_{x' \in E} \beta_{t+1}(x') A_{xx'} B_{x'y^{t+1}} \end{aligned}$$

**Зауваження.** При великих значеннях довжини ланцюга виникає потреба у шкалюванні [5, розділ 5] коефіцієнтів прямого та зворотного ходу, адже їхні значення стають нерозрізнювано малими для обчислювальних ресурсів. Процедура шкалювання полягає в наступному: на кожному кроці  $t$  після обчислення істинних змінних (1.4) слід виконати відповідне нормування

$$\forall x \in E : \hat{\alpha}_t(x) = \frac{\alpha_t(x)}{C_t}, \text{ де } C_t = \sum_{x' \in E} \alpha_t(x'),$$

а тоді коефіцієнти (1.5) нормуються так:

$$\forall x \in E : \hat{\beta}_t(x) = \frac{\beta_t(x)}{C_t}$$

## 1.4 Алгоритм Вітербі

Отримавши оптимальну модель  $\hat{\lambda} = (\hat{\pi}, \hat{A}, \hat{B})$  як розв'язок задачі навчання, перейдемо до так званої задачі декодування: віднайдемо ланцюжок прихованих станів системи. Алгоритм, який дозволяє ефективно розв'язати задачу декодування, називається алгоритмом Вітербі [5, розділ 6].

Отже, шукатимемо таку послідовність прихованих станів  $\hat{X}^1, \hat{X}^2, \dots, \hat{X}^T$ , яка найкращим чином описує наявні спостереження:

$$\hat{X} = \operatorname{argmax}_{x \in E^T} P(X = x | Y = y, \hat{\lambda}) = \operatorname{argmax}_{x \in E^T} P(X = x, Y = y | \hat{\lambda})$$

Введемо величини  $\delta_t(x)$  максимальної ймовірності спостереження ланцюжка довжини  $t$ , що закінчується станом  $x \in E$  в момент часу  $t$ :

$$\delta_t(x) = \max_{x^1, \dots, x^{t-1}} P(X^1 = x^1, \dots, X^{t-1} = x^{t-1}, X^t = x, Y^1 = y^1, \dots, Y^t = y^t | \hat{\lambda})$$



Вказані ймовірності можна визначити рекурентно:

$$\begin{aligned} t = 1 & \quad \forall x \in E : \delta_1(x) = \pi_x B_{xy^1} \\ t = \overline{2, T} & \quad \forall x \in E : \delta_t(x) = B_{xy^t} \cdot \max_{x' \in E} \left\{ \delta_{t-1}(x') A_{x'x} \right\} \end{aligned}$$

При цьому, щоб знайти оптимальний ланцюжок прихованих станів необхідно відстежувати аргумент, при якому досягається максимум  $\delta_t(x)$  для кожного  $t$  та  $x$ . Таким чином, алгоритм Вітербі знаходження найбільш імовірного ланцюжка прихованих станів є таким:

1) крок ініціалізації:

$$\forall x \in E : \delta_1(x) = \pi_x B_{xy^1}, \psi_1(x) = 1$$

2) обчислити коефіцієнти  $\delta_t(x)$  та відповідні аргументи  $\psi_t(x)$  :

$$\begin{aligned} \forall t = \overline{1, T}, \forall x \in E : & \quad \delta_t(x) = B_{xy^t} \cdot \max_{x' \in E} \left\{ \delta_{t-1}(x') A_{x'x} \right\} \\ \forall t = \overline{1, T}, \forall x \in E : & \quad \psi_t(x) = \operatorname{argmax}_{x' \in E} \left\{ \delta_{t-1}(x') A_{x'x} \right\} \end{aligned}$$

3) покласти зворотну точку відліку:

$$\begin{aligned} \hat{\delta} &= \max_{x \in E} \left\{ \delta_T(x) \right\} \\ \hat{\psi} &= \operatorname{argmax}_{x \in E} \left\{ \delta_T(x) \right\} \end{aligned}$$

4) визначити оптимальний ланцюжок станів (у зворотному порядку), починаючи з останнього  $\hat{x}^T = \hat{\psi}$  :

$$\forall t = \overline{T-1, 1} : \hat{x}^t = \psi_{t+1}(\hat{x}^{t+1})$$

## 1.5 Властивості точкових оцінок

Окреслимо основні інструменти математичної статистики, які будуть використані при побудові статистичних оцінок невідомих параметрів.

Вектор  $\vec{X} = (X_1, \dots, X_T)$  незалежних однаково розподілених випадкових величин з деякої параметричної сім'ї розподілів  $\mathfrak{F} = \{F_\theta(x), \theta \in \Theta\}$  називають вибіркою об'єму  $T$ . При цьому, параметр розподілу  $\theta$  може бути невідомим. Функція від вибірки  $S_T = S_T(\vec{X})$  називається статистикою.

Якщо значення статистики  $S_T(\vec{X})$  при заданій реалізації вибірки приймають за наближене значення невідомого параметра  $\theta$  розподілу  $F_\theta(x)$ , тоді  $S_T$  називають точковою оцінкою  $\theta$ .

**Означення 1.3.** Статистика  $S_T$  називається змістовною оцінкою  $\theta$ , якщо вона збігається за ймовірністю до істинного значення оцінюваного параметра, тобто

$$\forall \theta \in \Theta : S_T \xrightarrow[T \rightarrow \infty]{P} \theta \iff \forall \varepsilon > 0 : \lim_{T \rightarrow \infty} P(|S_T - \theta| \geq \varepsilon) = 0$$

**Означення 1.4.** Статистика  $S_T$  називається незміщеною оцінкою  $\theta$ , якщо її математичне сподівання дорівнює істинному значенню оцінюваного параметра, тобто

$$\forall \theta \in \Theta : M_\theta S_T = \theta$$

Також зазначимо для набору незалежних однаково розподілених випадкових величин так званий закон великих чисел [6]:

**Теорема 1.1.** Нехай  $\{X_t\}_{t \geq 1}$  — послідовність незалежних однаково розподілених випадкових величин зі скінченним математичним сподіванням  $MX_1 = m$ . Тоді випадкова величина  $\bar{X} = \frac{1}{T} \sum_{t=1}^T X_t$  збігається за ймовірністю до значення математичного сподівання  $m$ :

$$\bar{X} \xrightarrow[T \rightarrow \infty]{P} m \iff \forall \varepsilon > 0 : \lim_{T \rightarrow \infty} P\left(\left|\frac{1}{T} \sum_{t=1}^T X_t - m\right| \geq \varepsilon\right) = 0$$

## Висновки до розділу 1

У цьому розділі було розглянуто основний математичний апарат прихованих марковських моделей, необхідний для подальшого дослідження динаміки частково спостережуваного ланцюга Маркова на бінарних послідовностях

Використовуючи наведені викладки, проведемо у наступному розділі побудову оцінок для шуканих параметрів моделі.

## 2 ОЦІНКА ПАРАМЕТРІВ МОДЕЛІ

Структуруємо дослідження таким чином:

1) спершу формально опишемо досліджуваний об'єкт, визначимо ключові параметри системи та переконаємося, що утворена модель є марковською;

2) розв'яжемо задачу навчання: за наявними спостереженнями про динаміку набору функціоналів від станів прихованого ланцюга бінарних послідовностей оцінимо керуючий параметр системи, використовуючи математичний апарат прихованих марковських моделей та методи побудови статистичних оцінок;

3) розв'яжемо задачу декодування: за наявними спостереженнями та оцінкою керуючого параметра відновимо ланцюг прихованих станів;

4) розв'яжемо задачу локалізації: за відомими значеннями набору функціоналів від деякої невідомої підмножини стану прихованого ланцюга, оцінимо потужність та набір елементів цієї підмножини;

5) розв'яжемо задачу навчання, окреслену в пункті 2), враховуючи, що наявні спостереження є зашумленими, спотвореними.

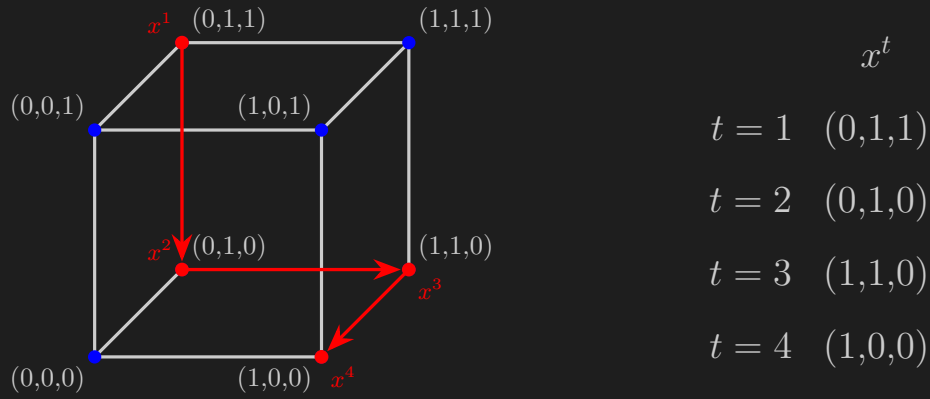
Кожен з підрозділів цього розділу матиме відповідну лаконічну назву: «Моделювання об'єкта дослідження», «Задача навчання», «Задача декодування», «Задача локалізації» та «Задача навчання за спотвореними спостереженнями».

### 2.1 Моделювання об'єкта дослідження

Розглянемо ланцюг Маркова  $\{X^t\}_{t=\overline{1,T}}$ , який приймає значення зі скінченної множини  $E = \{0,1\}^N$  — множини всеможливих бінарних послідовностей довжини  $N$ .

Динаміка ланцюга відбувається згідно з узагальненою моделлю Еренфестів: в кожен момент часу  $t$  навмання обирається число  $j$  з множини індексів  $\{1, 2, \dots, N\}$  бінарної послідовності  $X^t$  та відповідний елемент стану  $X_j^t$  залишається незмінним з імовірністю  $p$  або змінюється на протилежний бінарний символ з імовірністю  $1 - p$ .

Такого роду еволюцію бінарної послідовності довжини  $N$  можна уявити як так зване ліниве випадкове блукання вершинами  $N$ -вимірного куба. Наприклад, зобразимо на Рис. 2.1 реалізацію динаміки бінарних послідовностей довжини  $N = 3$  протягом  $T = 4$  моментів часу:



**Рисунок 2.1** – Реалізація динаміки ланцюга Маркова

Бачимо, що рух відбувається по ребрах від вершини до вершини. Для позначення сусідства між станом  $X^t$  та  $X^{t+1}$  введемо відстань Геммінга:

$$d_H(X^t, X^{t+1}) = \sum_{i=1}^N \mathbb{1}(X_i^t \neq X_i^{t+1})$$

В термінах цієї відстані перехід в одну з  $N$  сусідніх вершин відповідатиме значенню одиниці, а незмінність стану у послідовні моменти часу — значенню нуля. Внаслідок окресленої динаміки елементи

матриці перехідних імовірностей  $A$  ланцюга визначатимуться як

$$A_{xx'} = P(X^{t+1} = x' | X^t = x) = \begin{cases} p, & d_H(x, x') = 0 \\ \frac{1-p}{N}, & d_H(x, x') = 1 \\ 0, & \text{інакше} \end{cases} \quad (2.1)$$

Початковий розподіл  $\pi$  заданого ланцюга вважатимемо рівномірним на множині станів  $E$ , тобто

$$\forall x \in E : \pi_x = \frac{1}{2^N} \quad (2.2)$$

**Зауваження.** Рівномірний початковий розподіл  $\pi$  заданої марковської моделі  $\lambda = (\pi, A)$  є інваріантним через те, що матриця  $A$  є стохастичною та симетричною водночас. Іншими словами, виконується матрична рівність  $\pi A = \pi$  :

$$\forall x' \in E : \sum_{x \in E} \pi_x A_{xx'} = \frac{1}{2^N} \sum_{x \in E} A_{xx'} = \frac{1}{2^N} \sum_{x' \in E} A_{xx'} = \frac{1}{2^N} = \pi_{x'}$$

На додачу до ланцюга Маркова  $\{X^t\}_{t=\overline{1, T}}$  введемо послідовність випадкових величин  $\{Y^t\}_{t=\overline{1, T}}$ , які формуються таким чином:

$$Y^t = \left( Y_k^t \right)_{k=\overline{1, L}} = \left( \phi(X^t, I_k) \right)_{k=\overline{1, L}}, \quad t = \overline{1, T}, \quad (2.3)$$

де  $I = \{I_1, \dots, I_L\}$  — задані підмножини множини індексів  $\{1, 2, \dots, N\}$ , а функціонал  $\phi$  визначимо так:

$$\phi(X^t, I_k) = \sum_{i \in I_k} X_i^t \quad (2.4)$$

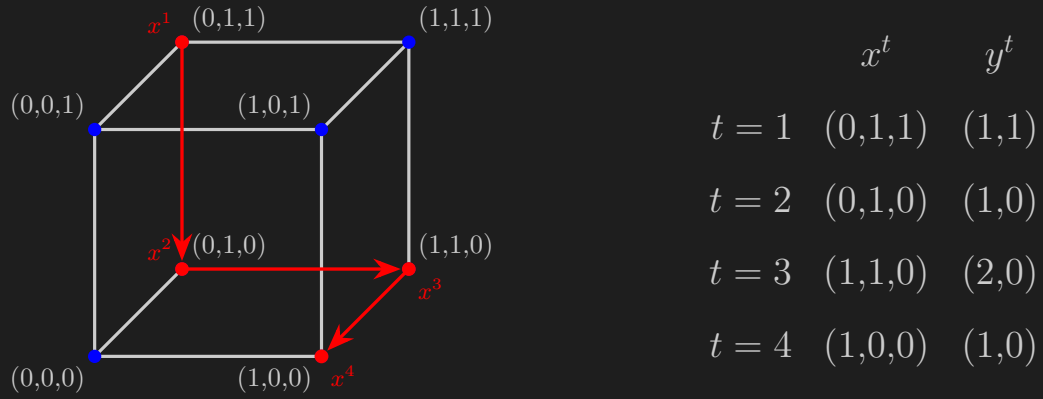
**Твердження 2.1.** *Послідовність  $\{(X^t, Y^t)\}_{t=\overline{1, T}}$  утворює приховану марковську модель  $\lambda = (\pi, A, B)$  з рівномірним початковим розподілом  $\pi$  (2.2), матрицею перехідних імовірностей (2.1) та*

матрицею  $B$ , елементи якої мають вигляд:

$$B_{xy} = P(Y^t = y | X^t = x) = \prod_{k=1}^L \mathbb{1} \left( y_k = \sum_{i \in I_k} x_i \right) \quad (2.5)$$

**Доведення.** За побудовою послідовності  $\{(X^t, Y^t)\}_{t=\overline{1, T}}$  означення прихованої марковської моделі (Озн. 1.2) виконується.  $\square$

Наприклад, наведемо на рисунку нижче реалізацію ланцюга Маркова (Рис. 2.1) для множини спостережуваних індексів вигляду  $I = \{I_1, I_2\} = \{(1, 2), (3)\}$  :



**Рисунок 2.2** – Реалізація динаміки прихованої марковської моделі

Бачимо, що умовні ймовірності спостережень при заданих прихованих станах визначаються як добуток індикаторів відповідності по кожній зі спостережуваних областей. До слова, підкреслимо, що множини  $I_1, \dots, I_L$  не обов'язково є покриттям чи розбиттям  $\{1, 2, \dots, N\}$ .

Таким чином, в результаті маємо сформовану приховану марковську модель  $\lambda = (\pi, A, B)$  з імовірністю  $p$  матриці  $A$  в ролі керуючого параметра системи.

## 2.2 Задача навчання

За наявними спостереженнями про динаміку набору функціоналів від станів прихованого ланцюга бінарних послідовностей оцінимо керуючий параметр  $p$  системи, використовуючи математичний апарат прихованих марковських моделей та методи побудови статистичних оцінок.

### 2.2.1 Ітераційний алгоритм Баума-Велша

В рамках ітераційного алгоритму Баума-Велша, максимізуємо функцію квазі-log правдоподібності (1.3). Враховуючи вигляд вектора початкового розподілу  $\pi$  (2.2), матриці перехідних ймовірностей  $A$  (2.1) та матриці  $B$  (2.5), функція правдоподібності (1.2) набуває вигляду:

$$\begin{aligned}
 P(Y = y | \lambda) &= \sum_{x \in E^T} L_\lambda = \sum_{x \in E^T} P(X = x, Y = y | \lambda) = \\
 &= \sum_{x \in E^T} \pi_{x^1} \cdot \prod_{t=1}^{T-1} A_{x^t x^{t+1}} \cdot \prod_{t=1}^T B_{x^t y^t} = \\
 &= \sum_{x \in E^T} \frac{1}{2^N} \cdot p^{\sum_{t=1}^{T-1} \mathbb{1}(d_H(x^t, x^{t+1})=0)} \cdot \left( \frac{1-p}{N} \right)^{\sum_{t=1}^{T-1} \mathbb{1}(d_H(x^t, x^{t+1})=1)} \cdot \mathbb{1}(x \in G),
 \end{aligned}$$

де добутки елементів матриці  $B$  позначені через множину  $G$  таким чином:

$$\mathbb{1}(x \in G) = \prod_{t=1}^T \prod_{k=1}^L \mathbb{1}\left(y_k = \sum_{i \in I_k} x_i\right)$$

Фактично, величина  $\mathbb{1}(x \in G)$  вказуватиме, чи є набір станів  $x \in E^T$  допустимим при заданій послідовності спостережень  $y$ . Подальші викладки будуть виконані у припущенні події  $\left\{ \mathbb{1}(x \in G) = 1 \right\}$ .



Відтак функція квазі-log правдоподібності матиме такий вигляд:

$$\begin{aligned}
 Q\left(\lambda^{(n)}, \lambda\right) &= \sum_{x \in E^T} L_{\lambda^{(n)}} \ln L_{\lambda} = \\
 &= \sum_{x \in E^T} L_{\lambda^{(n)}} \left[ \ln \frac{1}{2^N} + \ln p \cdot \sum_{t=1}^{T-1} \mathbb{1}\left(d_H\left(x^t, x^{t+1}\right) = 0\right) + \right. \\
 &\quad \left. + \ln \frac{1-p}{N} \cdot \sum_{t=1}^{T-1} \mathbb{1}\left(d_H\left(x^t, x^{t+1}\right) = 1\right) \right]
 \end{aligned}$$

Продовжуючи, домножимо на  $L_{\lambda^{(n)}} = P\left(X = x, Y = y \mid \lambda^{(n)}\right)$  кожен доданок:

$$\begin{aligned}
 Q\left(\lambda^{(n)}, \lambda\right) &= \ln \frac{1}{2^N} \cdot \sum_{x \in E^T} P\left(Y = y, X = x \mid \lambda^{(n)}\right) + \\
 &+ \ln p \cdot \sum_{t=1}^{T-1} \sum_{\substack{x, x' \in E \\ d_H(x, x')=0}} P\left(Y = y, X^t = x, X^{t+1} = x' \mid \lambda^{(n)}\right) + \\
 &+ \ln \frac{1-p}{N} \cdot \sum_{t=1}^{T-1} \sum_{\substack{x, x' \in E \\ d_H(x, x')=1}} P\left(Y = y, X^t = x, X^{t+1} = x' \mid \lambda^{(n)}\right)
 \end{aligned}$$

Зазначимо формулу переоцінки сумісної ймовірності подій  $U_1$  та  $U_2$  :

$$P\left(U_1 \mid U_2\right) \stackrel{\text{def}}{=} \frac{P\left(U_1, U_2\right)}{P\left(U_2\right)} \Rightarrow P\left(U_1, U_2\right) = P\left(U_1 \mid U_2\right) P\left(U_2\right) \quad (2.6)$$

Скористаємося формулою (2.6) для такого перетворення:

$$\begin{aligned}
 P_{\lambda^{(n)}}\left(Y = y, X^t = x, X^{t+1} = x'\right) &= \\
 &= P_{\lambda^{(n)}}\left(\underbrace{Y^1 = y^1, \dots, Y^t = y^t, X^t = x}_{U_1}, \underbrace{X^{t+1} = x'}_{U_2}, \right. \\
 &\quad \left. \underbrace{Y^{t+1} = y^{t+1}}_{U_3}, \underbrace{Y^{t+2} = y^{t+2}, \dots, Y^T = y^T}_{U_4}\right)
 \end{aligned}$$

Враховуючи введені позначення  $U_1, U_2, U_3, U_4$ , отримуємо:

$$\begin{aligned} P_{\lambda^{(n)}}(Y = y, X^t = x, X^{t+1} = x') &= P_{\lambda^{(n)}}(U_1, U_2, U_3, U_4) = \\ &= P_{\lambda^{(n)}}(U_1) P_{\lambda^{(n)}}(U_2 | U_1) P_{\lambda^{(n)}}(U_3 | U_1, U_2) P_{\lambda^{(n)}}(U_4 | U_1, U_2, U_3) \end{aligned}$$

До отриманих множників послідовно застосуємо марковську властивість (Озн. 1.1) та умовну незалежність спостережень при заданих прихованих станах (Озн. 1.2), щоб виразити шукану ймовірність через змінні прямого (1.4) і зворотного (1.5) ходу при поточному наближенні моделі  $\lambda^{(n)} = (\pi, A^{(n)}, B)$ :

$$P_{\lambda^{(n)}}(Y = y, X^t = x, X^{t+1} = x') = \alpha_t(x) A_{xx'}^{(n)} B_{x'y^{t+1}} \beta_{t+1}(x')$$

Тоді функція квазі-log правдоподібності набуде такого остаточного вигляду:

$$\begin{aligned} Q(\lambda^{(n)}, \lambda) &= \ln \frac{1}{2^N} \cdot \sum_{x \in E^T} P_{\lambda^{(n)}}(Y = y, X = x) + \\ &+ \ln p \cdot \sum_{t=1}^{T-1} \sum_{\substack{x, x' \in E \\ d_H(x, x')=0}} \alpha_t(x) A_{xx'}^{(n)} B_{x'y^{t+1}} \beta_{t+1}(x') + \\ &+ \ln \frac{1-p}{N} \cdot \sum_{t=1}^{T-1} \sum_{\substack{x, x' \in E \\ d_H(x, x')=1}} \alpha_t(x) A_{xx'}^{(n)} B_{x'y^{t+1}} \beta_{t+1}(x') \end{aligned}$$

Задача пошуку оптимального значення параметра  $p$  на кроці  $n+1$  при поточному наближенні  $p^{(n)}$  зводиться до наступного:

$$p^{(n+1)} = \operatorname{argmax}_p Q(p^{(n)}, p) \iff \frac{d}{dp} Q(p^{(n)}, p) = 0$$

Перш ніж переходити до пошуку екстремуму, вкажемо, що сума

$$\sum_{\substack{x, x' \in E \\ d_H(x, x')=0}} P_{\lambda^{(n)}} (Y = y, X^t = x, X^{t+1} = x') + \\ + \sum_{\substack{x, x' \in E \\ d_H(x, x')=1}} P_{\lambda^{(n)}} (Y = y, X^t = x, X^{t+1} = x')$$

еквівалентна

$$\sum_{x, x' \in E} P_{\lambda^{(n)}} (Y = y, X^t = x, X^{t+1} = x') \mathbb{1}(d_H(x, x') = 0) + \\ + \sum_{x, x' \in E} P_{\lambda^{(n)}} (Y = y, X^t = x, X^{t+1} = x') \mathbb{1}(d_H(x, x') = 1),$$

що, своєю чергою, спрощується як сума ймовірностей повної групи подій до виразу

$$\sum_{x \in E} P_{\lambda^{(n)}} (Y = y, X^t = x) = \sum_{x \in E} \alpha_t(x) \beta_t(x)$$

Отже, як розв'язок диференціального рівняння  $\frac{d}{dp} Q(p^{(n)}, p) = 0$  отримуємо таку ітераційну формулу переоцінки параметра  $p$ , починаючи з деякого початкового наближення  $p^{(0)}$  :

$$p^{(n+1)} = p^{(n)} \cdot \frac{\sum_{t=1}^{T-1} \sum_{x \in E} \alpha_t(x) B_{xy^{t+1}} \beta_{t+1}(x)}{\sum_{t=1}^{T-1} \sum_{x \in E} \alpha_t(x) \beta_t(x)} \quad (2.7)$$

Принагідно зазначимо, що при великих значеннях довжини ланцюга ( $T > 300$ ) виникає потреба у шкалюванні коефіцієнтів прямого та зворотного ходу згідно з процедурою, описаною на сторінці 15. Процедура нормування не вносить змін у вигляд ітераційної формули переоцінки параметра  $p$ .

### 2.2.2 Побудова оцінки методами математичної статистики

В межах побудованої прихованої марковської моделі  $\lambda = (\pi, A, B)$  при переоцінці параметра  $p$  за формулою (2.7) час виконання ітераційного алгоритму Баума-Велша суттєво підвищується при ускладненні моделі шляхом збільшення довжини ланцюга  $T$  чи довжини  $N$  стану ланцюга (мова йде про значення  $T > 1000$  та  $N > 8$ ).

Це пов'язано із двократним збільшенням потужності множини станів прихованого ланцюга  $E$  при збільшенні  $N$ , адже  $|E| = 2^N$ . Відтак багаторазово збільшуються і розміри відповідних матриць  $A$ ,  $B$  та вектора  $\pi$ .

Як наслідок, збільшується число доданків в операції кратного сумування безпосередньо у формулі переоцінки параметра (2.7).

Проте, враховуючи, що еволюція прихованого ланцюга Маркова відбувається згідно з узагальненою моделлю Еренфестів, матимемо змогу скористатися для розв'язку задачі оцінки невідомого параметра  $p$  статистичними методами обробки, за допомогою яких вдасться обійти обмеження на значення  $N$  та  $T$ .

Спершу визначимо, яка ймовірність того, що вектор спостережень  $Y^t$  при переході з деякого фіксованого моменту часу  $t$  до  $t + 1$  залишиться незмінним: або обраний елемент у відповідному прихованому стані  $X_j^t$  з індексом  $j$  з множини усіх спостережень  $I_1 \cup I_2 \cup \dots \cup I_L$  не змінив свого значення (це відбувається з імовірністю  $p$ ), або обраний індекс  $j$  взагалі лежить поза спостережуваною областю індексів:

$$P(Y^t = Y^{t+1}) = \frac{\left| \bigcup_{k=1}^L I_k \right|}{N} \cdot p + \frac{N - \left| \bigcup_{k=1}^L I_k \right|}{N} \quad (2.8)$$

Крім того, зважаючи на заданий спосіб еволюції моделі події рівності послідовних спостережень є незалежними, адже в послідовні моменти часу вибір елементу прихованого стану, який підлягатиме зміні, відбувається

навмання. Відтак  $\left\{ \mathbb{1}(Y^t = Y^{t+1}), t = \overline{1, T-1} \right\}$  — незалежні та однаково розподілені випадкові величини. А отже, з огляду на закон великих чисел (Теорема 1.1) має місце збіжність:

$$\frac{1}{T-1} \sum_{t=1}^{T-1} \mathbb{1}(Y^t = Y^{t+1}) \xrightarrow[T \rightarrow \infty]{P} M \mathbb{1}(Y^1 = Y^2) = P(Y^1 = Y^2) \quad (2.9)$$

З (2.8) та (2.9) випливає наступне твердження:

**Твердження 2.2.** *Змістовною і незміщеною оцінкою невідомого параметра  $p$  є статистика*

$$\hat{p} = 1 - \frac{N}{\left| \bigcup_{k=1}^L I_k \right|} \left( 1 - \frac{1}{T-1} \sum_{t=1}^{T-1} \mathbb{1}(Y^t = Y^{t+1}) \right) \quad (2.10)$$

Зауважимо, що формула (2.10) є значно простішою з обчислювальної точки зору, аніж ітераційний вираз (2.7).

### 2.3 Задача декодування

Використовуючи оцінене значення параметра  $\hat{p}$ , отримане в результаті застосування алгоритму навчання Баума-Велша або методів статистичної обробки, застосуємо алгоритм декодування Вітербі, кроки якого описані у розділі 1.4, для пошуку послідовності прихованих станів  $\hat{X}^1, \hat{X}^2, \dots, \hat{X}^T$ , яка найкращим чином описує наявні спостереження:

$$\hat{X} = \operatorname{argmax}_{x \in E^T} P(X = x \mid Y = y, \hat{p}) \quad (2.11)$$

### 2.4 Задача локалізації

Нехай окрім набору спостережень (2.3) протягом еволюції прихованої марковської моделі  $\lambda = (\pi, A, B)$  на кожному кроці  $t$  спостерігається деяке

додаткове значення  $Y_{I_*}^t$  функціонала (2.4) від прихованого стану ланцюга по деякій невідомій підмножині індексів  $I_* \subseteq \{1, 2, \dots, N\}$  :

$$Y_{I_*} = \left( Y_{I_*}^t \right)_{t=\overline{1, T}} = \left( \sum_{i \in I_*} X_i^t \right)_{t=\overline{1, T}}$$

Використовуючи результати задач навчання та декодування, спробуємо визначити, локалізувати за набором певних «сигналів» від прихованого ланцюга джерело їхнього надходження.

#### 2.4.1 Оцінка потужності шуканої множини

Перш за все, оцінимо кількість елементів множини  $I_*$ . В силу заданого способу еволюції моделі ймовірність незмінності сигналу у послідовні моменти часу можна визначити через керуючий параметр системи  $p$  аналогічно до міркувань стосовно ймовірності (2.8):

$$P(Y_{I_*}^t = Y_{I_*}^{t+1}) = \frac{|I_*|}{N} \cdot p + \frac{N - |I_*|}{N} \quad (2.12)$$

Отримана рівність дозволяє побудувати незміщену та змістовну оцінку для потужності  $|I_*|$ .

**Твердження 2.3.** *Змістовною і незміщеною оцінкою потужності множини  $I_*$  є статистика*

$$|\widehat{I_*}| = \frac{N}{1-p} \left( 1 - \frac{1}{T-1} \sum_{t=1}^{T-1} \mathbb{1}(Y_{I_*}^t = Y_{I_*}^{t+1}) \right) \quad (2.13)$$

**Доведення.** Зважаючи на вигляд ймовірності (2.12), переконуємося у незміщеності оцінки:

$$M|\widehat{I_*}| = M|I_*|$$

Оскільки динаміка моделі відбувається згідно з узагальненою моделлю Еренфестів, у послідовні моменти часу вибір елементу

прихованого стану, який підлягатиме зміні, відбувається навмання. Отже,  $\left\{ \mathbb{1}\left(Y_{I_*}^t = Y_{I_*}^{t+1}\right), t = \overline{1, T-1} \right\}$  є незалежними й однаково розподіленими випадковими величинами. А тоді згідно із законом великих чисел (Теорема 1.1) справедливо наступне:

$$\frac{1}{T-1} \sum_{t=1}^{T-1} \mathbb{1}\left(Y_{I_*}^t = Y_{I_*}^{t+1}\right) \xrightarrow[T \rightarrow \infty]{P} M \mathbb{1}\left(Y_{I_*}^1 = Y_{I_*}^2\right) = P\left(Y_{I_*}^1 = Y_{I_*}^2\right)$$

Використовуючи рівність (2.12), переконуємося у змістовності оцінки:

$$\widehat{|I_*|} \xrightarrow[T \rightarrow \infty]{P} |I_*|$$

□

Аналогічним чином побудуємо оцінку для потужності перетину множини  $I_*$  з індексами множин, які задають спостереження моделі. Вказана оцінка дозволить виявити взаємне розташування елементів множини неявних індексів та множини доступних для дослідження елементів прихованого стану ланцюга Маркова.

**Твердження 2.4.** *Нехай  $H \subseteq I_1 \cup I_2 \cup \dots \cup I_L$  — довільна підмножина множини спостережуваних індексів  $I_1 \cup I_2 \cup \dots \cup I_L$ . Тоді змістовною та незміщеною оцінкою потужності множини  $I_* \cap H$  є статистика*

$$\widehat{|I_* \cap H|} = \frac{N}{(T-1)(1-p)} \sum_{t=1}^{T-1} \mathbb{1}\left(Y_{I_*}^t \neq Y_{I_*}^{t+1}, Y_H^t \neq Y_H^{t+1}\right), \quad (2.14)$$

де позначено для довільного  $t = \overline{1, T-1}$ :

$$Y_H^t = \sum_{i \in H} X_i^t$$

**Доведення.** Одночасна зміна як сигналу, так і значення функціонала по множині  $H$  у послідовні моменти часу відбуватиметься лише тоді, коли обрано та змінено індекс прихованого стану, який

належить перетину  $I_* \cap H$  :

$$P(Y_{I_*}^t \neq Y_{I_*}^{t+1}, Y_H^t \neq Y_H^{t+1}) = \frac{|I_* \cap H|}{N} \cdot (1 - p)$$

Отже, переконуємося у незміщеності побудованої оцінки:

$$M|\widehat{I_* \cap H}| = M|I_* \cap H|$$

Крім того, оскільки індекс, який підлягатиме зміні в момент часу  $t$ , обирається навмання, набір  $\left\{ \mathbb{1}(Y_{I_*}^t \neq Y_{I_*}^{t+1}, Y_H^t \neq Y_H^{t+1}), t = \overline{1, T-1} \right\}$  утворює послідовність незалежних та однаково розподілених випадкових величин, для яких справедливий закон великих чисел (Теорема 1.1):

$$\frac{1}{T-1} \sum_{t=1}^{T-1} \mathbb{1}(Y_{I_*}^t \neq Y_{I_*}^{t+1}, Y_H^t \neq Y_H^{t+1}) \xrightarrow[T \rightarrow \infty]{P} P(Y_{I_*}^1 \neq Y_{I_*}^2, Y_H^1 \neq Y_H^2)$$

Відтак маємо змогу переконатися у змістовності оцінки:

$$|\widehat{I_* \cap H}| \xrightarrow[T \rightarrow \infty]{P} |I_* \cap H|$$

□

### 2.4.2 Оцінка шуканої множини

Стратегія визначення елементів, які безпосередньо входять в множину  $I_*$ , складатиметься з декількох кроків:

0) використовуючи оцінку керуючого параметра  $\hat{p}$ , обчислити потужність множини  $I_*$  через формулу (2.13);

1) із загальної множини індексів  $\{1, 2, \dots, N\}$  сформувати всеможливі підмножини довжиною  $|\widehat{I_*}|$ , тобто вибірку

$$\left\{ I_1, I_2, \dots, I_{C_N^{|\widehat{I_*}|}} \right\} \quad (2.15)$$

2) для кожного «кандидата»  $I_k$  з множини (2.15) згенерувати



послідовність значень функціонала (2.4) від декодованих прихованих станів (2.11) по відповідним індексам:

$$\hat{Y}_{I_k} = \left( \hat{Y}_{I_k}^t \right)_{t=\overline{1, T}} = \left( \sum_{i \in I_k} \hat{X}_i^t \right)_{t=\overline{1, T}}$$

3) за допомогою деякої заданої міри  $d$  оцінити для кожного  $I_k$  відстань між множиною  $\hat{Y}_{I_k}$  та сигналами  $Y_{I_*}$ ;

4) оцінкою  $\hat{I}$  множини  $I_*$  стане той «кандидат»  $I_k$  з множини (2.15), для якого відстань  $d$  буде найменшою:

$$\hat{I} = \underset{1 \leq k \leq C_N^{|\hat{I}_*|}}{\operatorname{argmin}} d \left( \hat{Y}_{I_k}, Y_{I_*} \right) \quad (2.16)$$

Міру близькості  $d$  між двома невід'ємними цілочисельними множинами однакової довжини  $\hat{Y}_{I_k}$  та  $Y_{I_*}$  визначатимемо або за допомогою квадратичної відстані

$$d_S \left( \hat{Y}_{I_k}, Y_{I_*} \right) = \sum_{t=1}^T \left( \hat{Y}_{I_k}^t - Y_{I_*}^t \right)^2, \quad (2.17)$$

або користуючись зваженою відстанню Жаккара [7]

$$d_J \left( \hat{Y}_{I_k}, Y_{I_*} \right) = 1 - \frac{\sum_{t=1}^T \min \left( \hat{Y}_{I_k}^t, Y_{I_*}^t \right)}{\sum_{t=1}^T \max \left( \hat{Y}_{I_k}^t, Y_{I_*}^t \right)} \quad (2.18)$$

Перевага зваженої відстані Жаккара полягає у її зосередженості на відрізку  $(0,1)$ , що дозволяє проводити додаткові аналогії міри між множинами як відсотка несхожості наборів. Крім того, на відміну від квадратичної відстані, відстань Жаккара оперує безпосередньо значеннями елементів вхідних множин, а не їхніми різницями.

## 2.5 Задача навчання за спотвореними спостереженнями

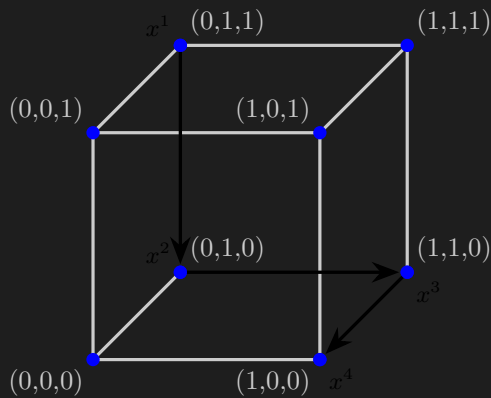
Припустимо, що для заданої прихованої марковської моделі  $\lambda = (\pi, A, B)$  значення функціонала (2.4) від прихованих станів ланцюга  $\{X^t\}_{t=\overline{1,T}}$  по множинах  $I_1, \dots, I_L$  спостерігаються із деякими ймовірностями спотворення  $q_1, q_2, \dots, q_L$  таким чином:

$$\phi(X^t, I_k) = \sum_{i \in I_k} \tilde{X}_i^t, \quad k = \overline{1, L}$$

де для  $i \in I_k$

$$\tilde{X}_i^t = \begin{cases} 1 - X_i^t, & \text{з імовірністю } q_k \\ X_i^t, & \text{з імовірністю } 1 - q_k \end{cases} \quad (2.19)$$

Наприклад, наведемо на рисунку нижче реалізацію прихованої марковської моделі (Рис. 2.2) при деяких ймовірностях спотворення  $q = (q_1, q_2)$  множин спостережуваних індексів  $I = \{I_1, I_2\} = \{(1,2), (3)\}$ . Червоним кольором позначені елементи, які змінили своє значення на протилежний бінарний символ:



	$x^t$	$\tilde{x}^t$	$y^t$
$t = 1$	(0,1,1)	( <b>1</b> ,1,1)	(2,1)
$t = 2$	(0,1,0)	(0,1, <b>1</b> )	(1,1)
$t = 3$	(1,1,0)	( <b>0</b> , <b>0</b> ,0)	(0,0)
$t = 4$	(1,0,0)	(1, <b>1</b> ,0)	(2,0)

**Рисунок 2.3** – Реалізація зашумленої динаміки прихованої марковської моделі

Переконаємося, що утворена «зашумлена» модель є прихованою марковською моделлю, аби мати змогу оцінити за наявними спотвореними даними керуючий параметр  $p$  та вектор імовірностей спотворень  $q = (q_1, q_2, \dots, q_L)$ , використовуючи ітераційний алгоритм навчання Баума-Велша.

Для кожного стану прихованого ланцюга Маркова  $\{X^t\}_{t=\overline{1,T}}$  в момент часу  $t$  введемо на множині спостережуваних областей  $I_1, \dots, I_L$  такі біноміально розподілені послідовності випадкових величин:

- 1)  $\left\{ \xi_{01}^k(X^t) \right\}_{k=\overline{1,L}}$  — кількості символів «0» прихованого стану  $X^t$ , які в силу можливості спотворення змінили своє значення на «1»;
- 2)  $\left\{ \xi_{11}^k(X^t) \right\}_{k=\overline{1,L}}$  — кількості символів «1» прихованого стану  $X^t$ , які не змінили свого значення.

Тоді значення функціоналів по множинах спостережень  $I_1, \dots, I_L$  в кожен момент часу  $t$  утворюватимуться як сума  $\left\{ \xi_{01}^k(X^t) + \xi_{11}^k(X^t) \right\}_{k=\overline{1,L}}$ , адже значущими для формування значень функціоналів є незалежні перетворення саме такого виду: «0»  $\xrightarrow{q_k}$  «1» та «1»  $\xrightarrow{1-q_k}$  «1», у чому можна переконатися на прикладі Рис. 2.3.

**Твердження 2.5.** *Якщо множини  $I_1, \dots, I_L$  є попарно неперетинними, то утворена послідовність  $\{(X^t, Y^t)\}_{t=\overline{1,T}}$  є прихованою марковською моделлю  $(\pi, A, B^q)$  з рівномірним початковим розподілом  $\pi$  (2.2), матрицею перехідних імовірностей (2.1) та матрицею  $B^q$ , елементи якої мають вигляд:*

$$B_{xy}^q = P(Y^t = y | X^t = x) = \prod_{k=1}^L P\left(\xi_{01}^k(x) + \xi_{11}^k(x) = y_k\right) \quad (2.20)$$

і для довільного  $k = \overline{1,L}$

$$\xi_{01}^k(x) \sim \text{Bin}\left(n_0(k), q_k\right), \quad \xi_{11}^k(x) \sim \text{Bin}\left(n_1(k), 1 - q_k\right) \quad (2.21)$$

є незалежними випадковими величинами, де позначено

$$\begin{aligned} n_0(k) &= |I_k| - \sum_{i \in I_k} x_i && \text{кількість «0» в множині} \\ & && I_k \text{ прихованого стану} \\ n_1(k) &= \sum_{i \in I_k} x_i && \text{кількість «1» в множині} \\ & && I_k \text{ прихованого стану} \end{aligned}$$

**Доведення.** За побудовою послідовності  $\{(X^t, Y^t)\}_{t=\overline{1, T}}$  означення прихованої марковської моделі (Озн. 1.2) виконується.  $\square$

Зауважимо, що як згортка суми двох незалежних біноміально розподілених випадкових величин компоненти матриці  $B^q$  обчислюються так:

$$\begin{aligned} B_{xy}^q &= \prod_{k=1}^L P\left(\xi_{01}^k(x) + \xi_{11}^k(x) = y_k\right) = \\ &= \prod_{k=1}^L \sum_{j=0}^{n_0(k)} P\left(\xi_{01}^k(x) = j\right) P\left(\xi_{11}^k(x) = y_k - j\right) = \\ &= \prod_{k=1}^L \sum_{j=0}^{n_0(k)} C_{n_0(k)}^j q_k^j (1 - q_k)^{n_0(k)-j} \cdot C_{n_1(k)}^{y_k-j} (1 - q_k)^{y_k-j} q_k^{n_1(k)-y_k+j} \end{aligned}$$

Наступним кроком, перш ніж переходити безпосередньо до розв'язку задачі навчання, враховуючи нотацію (2.19) введемо допоміжну послідовність  $\{Z^t\}_{t=\overline{1, T}}$  векторів, співрозмірних з довжиною бінарних послідовностей  $N$  :

$$\forall i \in \overline{1, N} : Z_i^t = \begin{cases} 1 \left( \tilde{X}_i^t \neq X_i^t \right), & i \in I_1 \cup I_2 \cup \dots \cup I_L \\ 0, & i \notin I_1 \cup I_2 \cup \dots \cup I_L \end{cases}$$

Таким чином, ця послідовність міститиме інформацію про спотвореність кожного зі спостережуваних елементів бінарної послідовності, що, своєю чергою, дозволить в алгоритмі Баума-Велша залучити у явному вигляді ймовірності спотворення  $q_1, \dots, q_L$ .

Тож, задавши деяке наближення  $\lambda^{(0)} = (\pi, A^{(0)}, B^{q^{(0)}})$ , покладемо для наступної ітерації  $n + 1$

$$\lambda^{(n+1)} = \operatorname{argmax}_{\lambda} Q\left(\lambda^{(n)}, \lambda\right) = \operatorname{argmax}_{\lambda} \sum_{x \in E^T} \sum_{z \in E^T} L_{\lambda^{(n)}} \ln L_{\lambda},$$

де функція повної правдоподібності матиме вигляд:

$$L_{\lambda} \equiv P(X = x, Z = z, Y = y | \lambda)$$

Відтак через формулу переоцінки сумісної ймовірності (2.6) матимемо:

$$\begin{aligned} L_{\lambda} &= P_{\lambda}(X = x, Z = z, Y = y) = \\ &= P_{\lambda}(X = x) P_{\lambda}(Z = z | X = x) P_{\lambda}(Y = y | Z = z, X = x) \end{aligned}$$

Почергово розглянемо кожен множник. З огляду на формулу (1.1) та вигляд матриці  $A$  й вектора  $\pi$ , скінченновимірний розподіл

$$P_{\lambda}(X = x) = \frac{1}{2^N} \cdot p^{\sum_{t=1}^{T-1} \mathbb{1}(d_H(x^t, x^{t+1})=0)} \cdot \left(\frac{1-p}{N}\right)^{\sum_{t=1}^{T-1} \mathbb{1}(d_H(x^t, x^{t+1})=1)}$$

Як наслідок марковської властивості (Озн. 1.1) та покоординатної умовної незалежності

$$\begin{aligned} P_{\lambda}(Z = z | X = x) &= \prod_{t=1}^T P_{\lambda}(Z^t = z^t | X^t = x^t) = \\ &= \prod_{t=1}^T \prod_{i=1}^N P_{\lambda}(Z_i^t = z_i^t | X_i^t = x_i^t) = \\ &= \prod_{t=1}^T \prod_{i=1}^N \prod_{k=1}^L \left( q_k^{\mathbb{1}(z_i^t=1)} \cdot (1 - q_k)^{\mathbb{1}(z_i^t=0)} \right)^{\mathbb{1}(i \in I_k)} \end{aligned}$$

Наостанок, застосовуючи на додачу до марковської властивості умову неперетинності множин  $I_1, \dots, I_L$ , матимемо

$$\begin{aligned} P_\lambda(Y = y \mid Z = z, X = x) &= \prod_{t=1}^T P_\lambda(Y^t = y^t \mid Z^t = z^t, X^t = x^t) = \\ &= \prod_{t=1}^T \prod_{k=1}^L P_\lambda(Y_k^t = y_k^t \mid Z^t = z^t, X^t = x^t) = \\ &= \prod_{t=1}^T \prod_{k=1}^L \mathbb{1} \left( y_k^t = \sum_{i \in I_k} x_i^t \mathbb{1}(z_i^t=0) \cdot (1 - x_i^t)^{\mathbb{1}(z_i^t=1)} \right), \end{aligned}$$

що вказує, чи є набір станів  $x \in E^T$  та послідовність індикаторів спотворення  $z \in E^t$  допустимими при заданій послідовності спостережень  $y$ . Надалі припускатимемо достовірність цієї події. В результаті отримуємо:

$$\begin{aligned} L_\lambda &= P_\lambda(X = x, Z = z, Y = y) = \\ &= \frac{1}{2^N} \cdot p^{\sum_{t=1}^{T-1} \mathbb{1}(d_H(x^t, x^{t+1})=0)} \cdot \left( \frac{1-p}{N} \right)^{\sum_{t=1}^{T-1} \mathbb{1}(d_H(x^t, x^{t+1})=1)} \times \\ &\quad \times \prod_{t=1}^T \prod_{i=1}^N \prod_{k=1}^L \left( q_k^{\mathbb{1}(z_i^t=1)} \cdot (1 - q_k)^{\mathbb{1}(z_i^t=0)} \right)^{\mathbb{1}(i \in I_k)} \end{aligned}$$

Домноживши на  $L_{\lambda^{(n)}} = P_{\lambda^{(n)}}(X = x, Z = z, Y = y)$  отриманий вираз, функція квазі-log правдоподібності розписуватиметься як

$$\begin{aligned} Q(\lambda^{(n)}, \lambda) &= \sum_{x \in E^T} \sum_{z \in E^T} L_{\lambda^{(n)}} \ln L_\lambda = \\ &= \ln \frac{1}{2^N} \cdot \sum_{x \in E^T} \sum_{z \in E^T} P_{\lambda^{(n)}}(X = x, Z = z, Y = y) + \\ &+ \ln p \cdot \sum_{t=1}^{T-1} \sum_{x \in E^T} \sum_{z \in E^T} P_{\lambda^{(n)}}(X = x, Z = z, Y = y) \mathbb{1}(d_H(x^t, x^{t+1}) = 0) + \\ &+ \ln \frac{1-p}{N} \cdot \sum_{t=1}^{T-1} \sum_{x \in E^T} \sum_{z \in E^T} P_{\lambda^{(n)}}(X = x, Z = z, Y = y) \mathbb{1}(d_H(x^t, x^{t+1}) = 1) + \end{aligned}$$

$$\begin{aligned}
& + \sum_{t=1}^T \sum_{i=1}^N \sum_{k=1}^L \ln q_k \sum_{x \in E^T} \sum_{z \in E^T} P_{\lambda^{(n)}}(X = x, Z = z, Y = y) \mathbb{1}\left(z_i^t = 1, i \in I_k\right) + \\
& + \sum_{t=1}^T \sum_{i=1}^N \sum_{k=1}^L \ln(1 - q_k) \sum_{x \in E^T} \sum_{z \in E^T} P_{\lambda^{(n)}}(X = x, Z = z, Y = y) \mathbb{1}\left(z_i^t = 0, i \in I_k\right)
\end{aligned}$$

Тоді аналогічно до викладок на сторінці 25 через змінні прямого (1.4) та зворотного (1.5) ходу при поточному наближенні моделі  $\lambda^{(n)} = (\pi, A^{(n)}, B^{q^{(n)}})$  матимемо:

$$\begin{aligned}
Q\left(\lambda^{(n)}, \lambda\right) &= \ln \frac{1}{2^N} \cdot \sum_{x \in E^T} \sum_{z \in E^T} P_{\lambda^{(n)}}(X = x, Z = z, Y = y) + \\
& + \ln p \cdot \sum_{t=1}^{T-1} \sum_{\substack{x, x' \in E \\ d_H(x, x')=0}} \alpha_t(x) A_{xx'}^{(n)} B_{x'y^{t+1}}^{q^{(n)}} \beta_{t+1}(x') + \\
& + \ln \frac{1-p}{N} \cdot \sum_{t=1}^{T-1} \sum_{\substack{x, x' \in E \\ d_H(x, x')=1}} \alpha_t(x) A_{xx'}^{(n)} B_{x'y^{t+1}}^{q^{(n)}} \beta_{t+1}(x') + \\
& + \sum_{t=1}^T \sum_{k=1}^L \ln q_k \sum_{x \in E} \sum_{x' \in E} \alpha_{t-1}(x') A_{x'x}^{(n)} q_k^{(n)} \beta_t(x) \sum_{i \in I_k} P(y^t | x, z_i = 0) + \\
& + \sum_{t=1}^T \sum_{k=1}^L \ln(1 - q_k) \sum_{x \in E} \sum_{x' \in E} \alpha_{t-1}(x') A_{x'x}^{(n)} (1 - q_k^{(n)}) \beta_t(x) \sum_{i \in I_k} P(y^t | x, z_i = 1),
\end{aligned}$$

де події  $\{z_i = 0\}$  чи  $\{z_i = 1\}$  передають інформацію щодо того, чи є спотвореним для поточного стану  $x \in E$  елемент  $x_i$  з індексом  $i \in I_k$ . А відтак в силу природи спостережень

$$\begin{aligned}
P_{x,i,0}^{q^{(n)}} &\equiv P(Y^t = y^t | X^t = x, z_i = 0) = \\
&= P\left(\widetilde{\xi}_{01}^k(x) + \widetilde{\xi}_{11}^k(x) = y_k - x_i\right) \cdot \prod_{\substack{m=1, \overline{L} \\ m \neq k}} P\left(\xi_{01}^m(x) + \xi_{11}^m(x) = y_m\right)
\end{aligned}$$

та відповідно

$$P_{x,i,1}^{q^{(n)}} \equiv P(Y^t = y^t \mid X^t = x, z_i = 1) = \\ = P\left(\widetilde{\xi}_{01}^k(x) + \widetilde{\xi}_{11}^k(x) = y_k - (1 - x_i)\right) \cdot \prod_{\substack{m=\overline{1,L} \\ m \neq k}} P\left(\xi_{01}^m(x) + \xi_{11}^m(x) = y_m\right),$$

де аналогічно до випадкових величин (2.21) величини

$$\widetilde{\xi}_{01}^k(x) \sim \text{Bin}(\widetilde{n}_0(k), q_k), \quad \widetilde{\xi}_{11}^k(x) \sim \text{Bin}(\widetilde{n}_1(k), 1 - q_k)$$

є незалежними випадковими величинами з параметрами

$$\widetilde{n}_0(k) = |I_k| - 1 - \sum_{j \in I_k \setminus \{i\}} x_j$$

кількість «0» в множині  $I_k$   
прихованого стану при відомій  
інформації про спотворення  $x_i$

$$\widetilde{n}_1(k) = \sum_{j \in I_k \setminus \{i\}} x_j$$

кількість «1» в множині  $I_k$   
прихованого стану при відомій  
інформації про спотворення  $x_i$

Як наслідок, складові ймовірностей  $P_{x,i,0}^{q^{(n)}}$  та  $P_{x,i,1}^{q^{(n)}}$  обчислюватимуться як згортки розподілів незалежних біноміально розподілених випадкових величин.

Отже, знайдемо екстремуми отриманої функції квазі-log правдоподібності як розв'язки рівнянь

$$\frac{\partial}{\partial p} Q(\lambda^{(n)}, \lambda) = 0 \quad \text{та} \quad \frac{\partial}{\partial q_k} Q(\lambda^{(n)}, \lambda) = 0, \quad k = \overline{1, L}$$

Враховуючи аналогічні (стр. 27) кроки спрощення знаменників отриманих виразів, в результаті отримуємо такі ітераційні формули переоцінки параметра  $p$  та компонент вектора  $q$ , починаючи з деякого



початкового наближення  $p^{(0)}$  та  $q^{(0)}$  :

$$p^{(n+1)} = p^{(n)} \cdot \frac{\sum_{t=1}^{T-1} \sum_{x \in E} \alpha_t(x) B_{xy^{t+1}}^{q^{(n)}} \beta_{t+1}(x)}{\sum_{t=1}^{T-1} \sum_{x \in E} \alpha_t(x) \beta_t(x)} \quad (2.22)$$

$$q_k^{(n+1)} = q_k^{(n)} \cdot \frac{\sum_{t=1}^T \sum_{x \in E} \sum_{x' \in E} \alpha_{t-1}(x') A_{x'x}^{(n)} \beta_t(x) \sum_{i \in I_k} P_{x,i,1}^{q^{(n)}}}{|I_k| \sum_{t=1}^T \sum_{x \in E} \alpha_t(x) \beta_t(x)}, \quad k = \overline{1, L} \quad (2.23)$$

Знову ж таки, що при великих значеннях довжини ланцюга ( $T > 300$ ) виникає потреба у шкалюванні коефіцієнтів прямого та зворотного ходу згідно з процедурою, описаною на сторінці 15. Процедура нормування не вносить змін у вигляд ітераційних формул переоцінки параметра  $p$  чи компонент вектора  $q$ .

## Висновки до розділу 2

В рамках дослідження різних характеристик моделі за наявними спостереженнями про динаміку набору функціоналів від станів прихованого ланцюга бінарних послідовностей вдалося розв'язати декілька задач, а саме: побудовано оцінку керуючого параметра системи; відтворено послідовність прихованих станів; локалізовано джерело надходження значень набору функціоналів від деякої невідомої підмножини стану прихованого ланцюга; побудовано оцінку керуючого параметра системи, враховуючи зашумленість спостережуваних даних.

Надалі згенеруємо модель та проведемо чисельний експеримент, на основі якого перевіримо ефективність отриманих аналітичних результатів.

### 3 ПРОВЕДЕННЯ ЧИСЕЛЬНОГО ЕКСПЕРИМЕНТУ

Для імплементації алгоритмів розв’язування задачі побудови оцінок невідомих параметрів моделі було використано засоби мови програмування Python версії 3.8.10 в інтегрованому середовищі розробки Visual Studio Code версії 1.78.2.

Вибір мови програмування зумовлювався широким арсеналом вбудованих програмних пакетів мови Python для роботи з масивами даних та математичними обчисленнями (бібліотеки NumPy, itertools, SciPy, random, numda), а також наявними інструментами для візуалізації даних (пакети pandas, matplotlib). Додаток Б містить тексти ключових блоків коду, необхідних для проведення чисельного експерименту.

Для ефективного керування великою кількістю взаємопов’язаних програмних блоків (функцій), а також для більш наочної демонстрації отриманих результатів було розроблено графічний інтерфейс користувача засобами пакета PySimpleGUI мови Python. Додаток А містить опис та приклад роботи розробленого програмного модуля.

#### 3.1 Моделювання об’єкта дослідження

Для експериментальної перевірки ефективності використаних методів необхідно згенерувати набір бінарних послідовностей довжини  $N$  із заданим параметром  $p$ , при цьому еволюція ланцюга має відбуватися згідно з узагальненою моделлю Еренфестів.

Тож нехай  $x^1 \in \{0,1\}^N$  — детермінований або обраний навмання на множині станів  $E$  початковий стан. Тоді подальше генерування ланцюга від стану  $x^t$  до стану  $x^{t+1}$  протягом  $T$  моментів часу при заданому параметрі  $p$  відбуватиметься за таким алгоритмом:

- 1) обрати навмання для стану  $x^t$  індекс  $j \in \{1, 2, \dots, N\}$ ;
- 2) згенерувати  $u^t \sim U(0,1)$  — випадкову величину  $u^t$  з рівномірного на відріжку  $(0,1)$  розподілу;
- 3) розглянути дві альтернативи:
  - а) якщо  $u^t \leq p$ , то всі елементи наступного стану  $x^{t+1}$  покласти рівними елементам поточного стану  $x^t$  :

$$\forall i = \overline{1, N} : x_i^{t+1} = x_i^t$$

- б) якщо  $u^t > p$ , то елемент  $x_j^{t+1}$  наступного стану  $x^{t+1}$  покласти протилежним до елементу  $x_j^t$  бінарним символом, а решту значень вважати рівними елементам поточного стану  $x^t$  :

$$\begin{aligned} \forall i \neq j : x_i^{t+1} &= x_i^t \\ x_j^{t+1} &= 1 - x_j^t \end{aligned}$$

- 4) повторити кроки 1) – 3) до формування ланцюга довжиною  $T$ .

Таким чином, у рамках чисельного експерименту було згенеровано прихований ланцюг Маркова протягом  $T = 200$  моментів часу для бінарних послідовностей довжини  $N = 5$  при заданому параметрі моделі  $p = 0.2$ , починаючи з початкового стану  $x^1 = (0,0,0,1,0)$ .

Значення спостережень формувалися як суми значень елементів прихованого стану по таким неперетинним множинам індексів:

$$I = \{I_1, I_2\} = \{(2,3), (1,4)\} \quad (3.1)$$

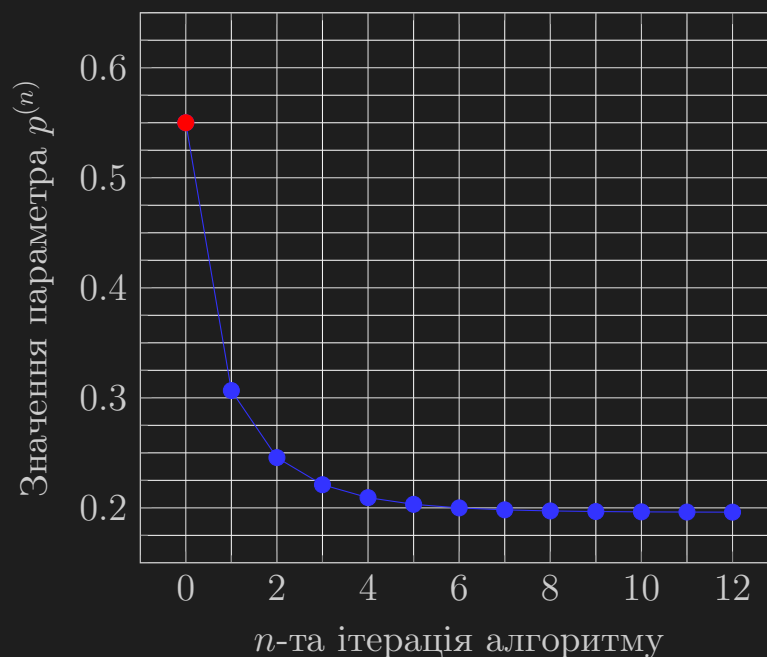
Отже, задано всі необхідні ввідні дані для того, щоб визначити приховану марковську модель  $\lambda = (\pi, A, B)$  : початковий розподіл  $\pi$  (2.2), матрицю перехідних імовірностей (2.1) та матрицю  $B$  (2.5).

## 3.2 Задача навчання

За наявними спостереженнями про динаміку набору функціоналів від станів прихованого ланцюга бінарних послідовностей оцінимо керуючий параметр системи та порівняємо отриману величину з істинним значенням параметра.

### 3.2.1 Ітераційний алгоритм Баума-Велша

Використовуючи формулу переоцінки (2.7), продемонструємо на Рис. 3.1 збіжність алгоритму Баума-Велша при оцінці параметра  $p$ , починаючи з наближення  $p^{(0)} = 0.55$ .



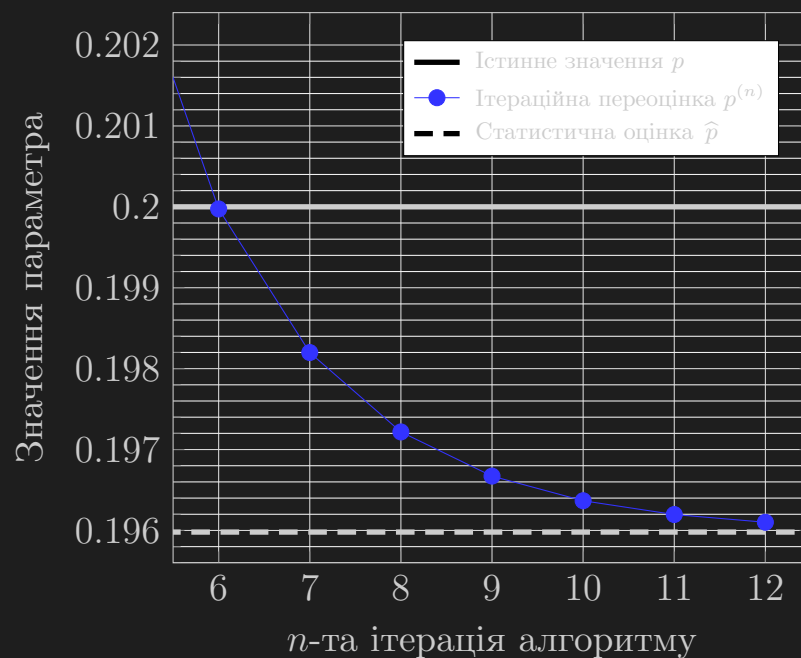
**Рисунок 3.1** – Ітерації алгоритму Баума-Велша при оцінці параметра  $p$

За  $n = 12$  ітерацій алгоритм досягає точності  $\varepsilon = 0.0001$  переоцінки оцінюваного параметра. При цьому, отримане значення  $p^{(12)} = 0.1959$  відрізняється від свого істинного значення  $p = 0.2$  на величину  $\delta = 0.0041$ .

### 3.2.2 Побудова оцінки методами математичної статистики

Застосуємо інший підхід для обчислення значення керуючого параметра моделі за наявними спостереженнями: знайдемо точкову оцінку  $\hat{p}$  згідно з виразом (2.10).

Отримана статистична оцінка відрізняється від ітераційної оцінки на величину, що складає менш ніж  $\delta = p^{(12)} - \hat{p} = 1.185 \cdot 10^{-6}$ . Інакше кажучи, для досліджуваної прихованої марковської моделі оцінка за ітераційним алгоритмом збігається до значення змістовної статистичної оцінки, яка, своєю чергою, є безпосереднім відгуком спостережуваних даних. Окреслений взаємозв'язок проілюстровано на рисунку нижче:



**Рисунок 3.2** – Збіжність оцінки ітераційного алгоритму до значення змістовної статистичної оцінки

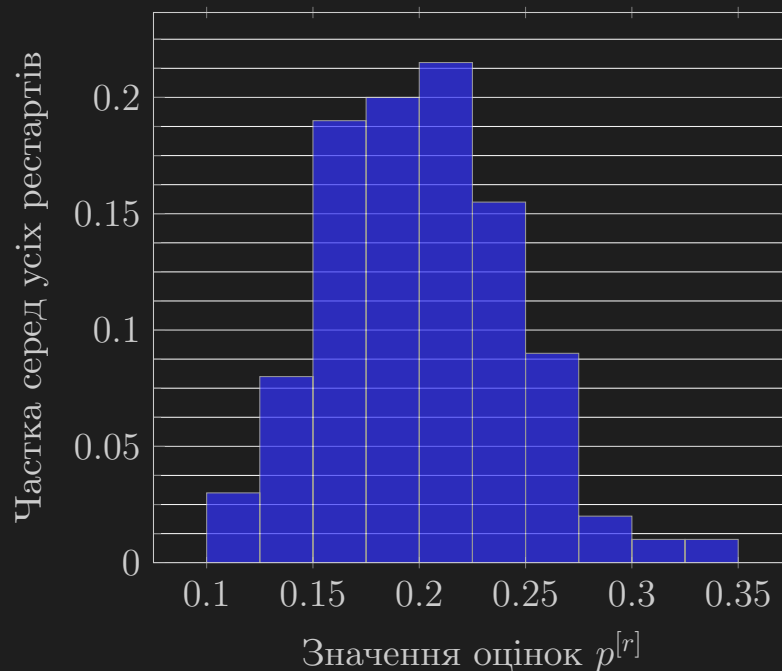
Отже, при збільшенні довжини  $T$  спостережуваного ланцюга ймовірність відхилення оцінки за алгоритмом Баума-Велша від істинного значення параметра буде прямувати до нуля.

### 3.2.3 Висновки щодо точності оцінювання

Оскільки ітераційний алгоритм Баума-Велша є збіжним лише до точки локального максимуму функції правдоподібності [1, розділ 15], виконаємо висновки щодо ефективності використаних методів для розв'язку задачі навчання шляхом аналізу результатів  $R = 200$  незалежних запусків алгоритму оцінювання невідомого параметра.

А саме: щоразу генеруватимемо ланцюг Маркова з параметром  $p = 0.2$  і множиною спостережуваних індексів (3.1) з навмання обраного стану  $x^1 \in E$ . Крім того, розпочинатимемо черговий ітераційний процес переоцінки з наближення  $p^{(0)} \in (0.05, 0.95)$ , обраного навмання.

В результаті було отримано вибірку оцінок  $\{p^{[r]}\}_{r=1, \overline{R}}$ , гістограма яких наведена нижче:



**Рисунок 3.3** – Гістограма значень оцінок  $R = 200$  незалежних рестартів алгоритму Баума-Велша

Вибіркове середнє сформованої вибірки складає

$$\mu_p = \frac{1}{R} \sum_{r=1}^R p^{[r]} = 0.2004,$$

а вибіркова дисперсія

$$S_p^2 = \frac{1}{R-1} \sum_{r=1}^R \left( p^{[r]} - \mu_p \right)^2 = 0.0019 \quad (3.2)$$

Отже, ітераційний алгоритм Баума-Велша продемонстрував високу точність оцінювання:

$$\delta_p = |p - \mu_p| = |0.2 - 0.2004| = 0.0004 \quad (3.3)$$

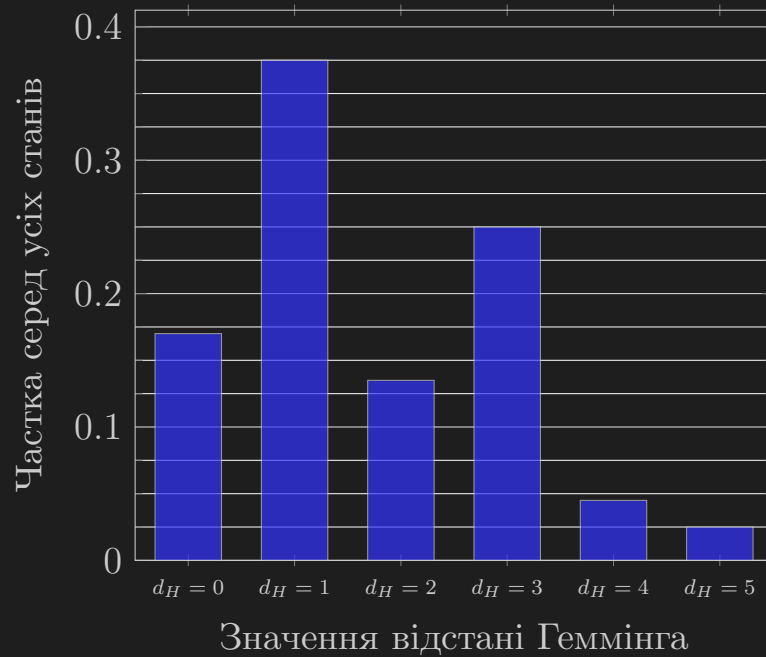
### 3.3 Задача декодування

Відтворимо ланцюг бінарних послідовностей згенерованої у розділі 3.1 моделі шляхом застосування алгоритму декодування Вітербі. Для цього використаємо оцінене значення керуючого параметра  $p^{(12)} \approx \hat{p} = 0.1959$ , отримане через формулу переоцінки (2.7) або шляхом побудови точкової оцінки (2.10).

Якість отриманих результатів охарактеризуємо через порівняння в кожен момент часу  $t$  істинної прихованої бінарної послідовності  $X^t$  та декодованої  $\hat{X}^t$  за допомогою відстані Геммінга:

$$d_H(X^t, \hat{X}^t) = \sum_{i=1}^N \mathbb{1}(X_i^t \neq \hat{X}_i^t)$$

Таким чином, чим більше символів між справжнім та декодованим станами збігатимуться, тим меншою буде відповідна відстань Геммінга.



**Рисунок 3.4** – Результати алгоритму декодування Вітербі

З гістограми результатів (Рис. 3.4) видно, що 17% усього ланцюга декодовано правильно. Наявність близько 40% помилок в одному символі може бути наслідком того, що одного елемента стану немає серед спостережуваних областей (3.1) ланцюга. Крім того, оцінка керуючого параметра має похибку  $\delta = 0.0041$  відносно свого істинного значення, що також впливає на результати задачі декодування.

Наступним кроком для кожного незалежного рестарту, проведеного у розділі 3.2.3 (Рис. 3.3), було виконано алгоритм декодування прихованих станів та отримано  $R = 200$  гістограм виду Рис. 3.4. Отримані розподіли значень для кожної з відстаней Геммінга наведені у Табл. 3.1.

**Таблиця 3.1** – Розподіли відстаней Геммінга для  $R = 200$  рестартів алгоритму декодування

	$d_H = 0$	$d_H = 1$	$d_H = 2$	$d_H = 3$	$d_H = 4$	$d_H = 5$
$\mu_{d_H}$	28.25	28.02	18.6	18.95	3.04	3.16
$S_{d_H}^2$	36.55	35.67	17.65	21.16	2.68	2.86



Як підсумок, точність результатів помітно варіюється від рестарту до рестарту. Причому закономірності вищої ефективності алгоритму декодування при точнішому значенні оцінки керуючого параметра як відносно свого істинного значення, так і відносно своєї змістовної точкової оцінки не прослідковується.

### 3.4 Задача локалізації

Значення додаткових спостережуваних «сигналів» було згенеровано як суми елементів прихованого стану по множині індексів  $I_* = (1,3,5)$ . Маючи з розділів 3.2 та 3.3 оцінку керуючого параметра та найбільш імовірний відтворений ланцюг бінарних послідовностей, віднайдемо набір елементів множини  $I_*$ .

Перш за все, з Табл. 3.2 робимо висновок, що кількості спостережень  $T = 200$  недостатньо для отримання точної оцінки потужності множини неявних індексів за формулою (2.13).

**Таблиця 3.2** – Залежність значення змістовної оцінки потужності  $|\widehat{I_*}|$  від довжини ланцюга  $T$

$T$	200	400	600	800	1000
$\hat{p}$	0.1959	0.1823	0.1882	0.2099	0.2092
$ \widehat{I_*} $	2	2	2	3	3

Однак, оскільки довжина бінарних послідовностей  $N$  є невеликою, для оцінки потужності шуканої множини можна скористатися емпіричною формулою вигляду

$$|\widehat{I_*}| = \max_{1 \leq t \leq 200} y_{I_*}^t = 3$$

Застосуємо отримане значення до виразу (2.16), щоб віднайти елементи, які безпосередньо входять в  $I_*$  : квадратична відстань (2.17) вказуватиме на сукупність індексів  $\hat{I}_S = (1,2,5)$ , а зважена відстань Жаккара (2.18) — на сукупність  $\hat{I}_J = (1,2,3)$ .

Дилему можна вирішити шляхом збільшення  $T$  та подальшого використання змістовної оцінки (2.14) для визначення взаємного розташування елементів множини неявних індексів відносно спостережуваних індексів (3.1).

### 3.5 Задача навчання за спотвореними спостереженнями

Для кожної із спостережуваних областей (3.1) змодельованого у розділі 3.1 ланцюга  $x^1, \dots, x^T$  було обрано такі ймовірності спотворення:  $q = (q_1, q_2) = (0.05, 0.1)$ .

Окреслимо алгоритм утворення зашумленого ланцюга  $\tilde{x}^1, \dots, \tilde{x}^T$  із заданими коефіцієнтами  $q_1$  та  $q_2$  :

1) для кожного з індексів  $i \in I_k$ ,  $k = \overline{1, L}$  поточного стану  $x^t$  згенерувати рівномірно розподілену випадкову величину  $u_i \sim U(0,1)$  :

а) якщо  $u_i \leq q_k$ , то покласти

$$\tilde{x}_i^t = 1 - x_i^t$$

б) якщо  $u_i > q_k$ , то покласти

$$\tilde{x}_i^t = x_i^t$$

2) для тих індексів  $i \notin \bigcup_{k=1}^L I_k$ , які не належать жодній із множин спостережуваних індексів покласти

$$\tilde{x}_i^t = x_i^t$$

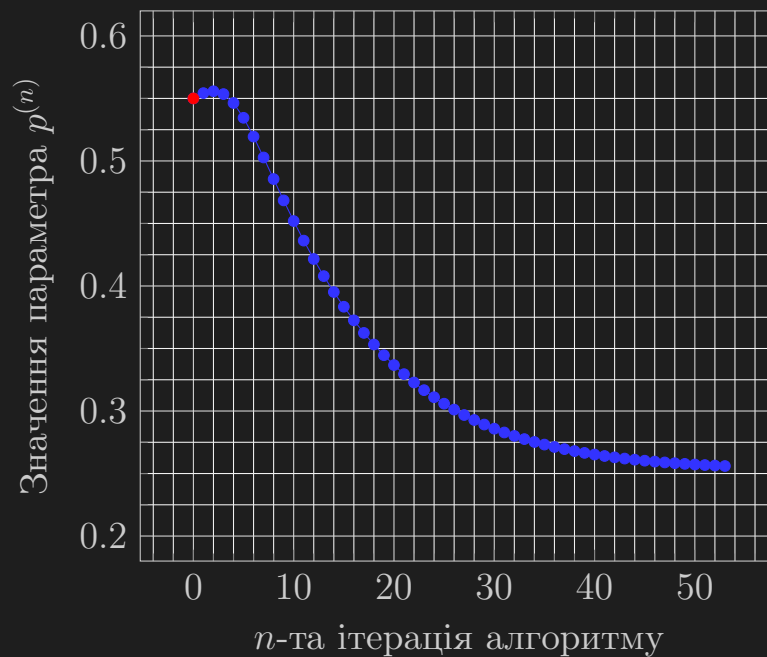
3) повторити кроки 1) – 2) для кожного стану  $x^t$ ,  $t = \overline{1, T}$ .

Таким чином, маємо змогу визначити приховану марковську модель  $\lambda = (\pi, A, B^q)$  з початковим розподілом  $\pi$  (2.2), матрицею перехідних імовірностей (2.1) та матрицею умовних імовірностей спостережень при заданих прихованих станах  $B^q$  (2.20).

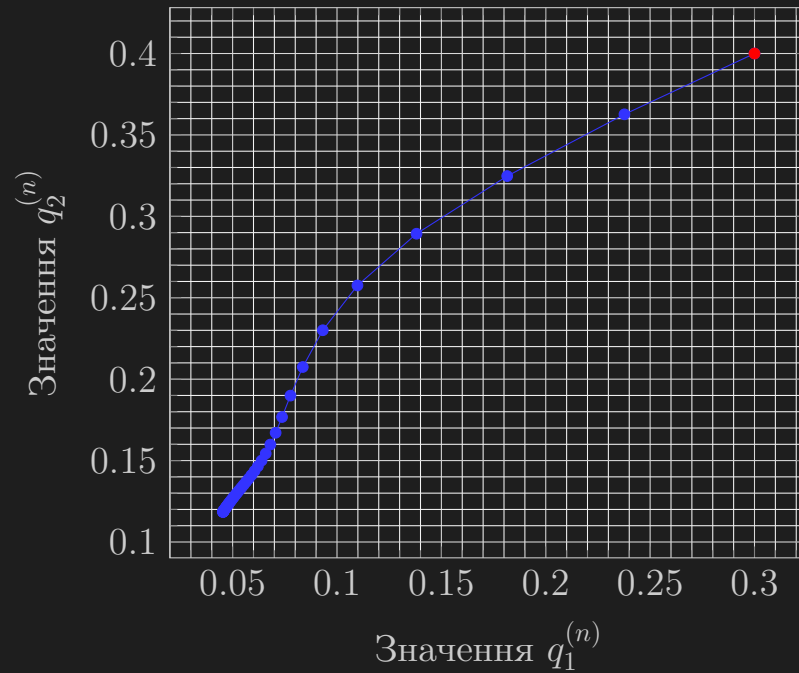
### 3.5.1 Ітераційний алгоритм Баума-Велша

Використовуючи формули переоцінки (2.22) та (2.23), оцінимо керуючий параметр системи та ймовірності викривлення за наявними споствореними спостереженнями про динаміку набору функціоналів від станів прихованого ланцюга бінарних послідовностей.

Рис. 3.5 та Рис. 3.6 демонструють результати ітераційної переоцінки невідомих параметрів моделі при початкових наближеннях  $p^{(0)} = 0.55$  й  $q^{(0)} = (0.3, 0.4)$ .



**Рисунок 3.5** – Ітерації алгоритму Баума-Велша при оцінці параметра  $p$ , враховуючи спотвореність спостережень



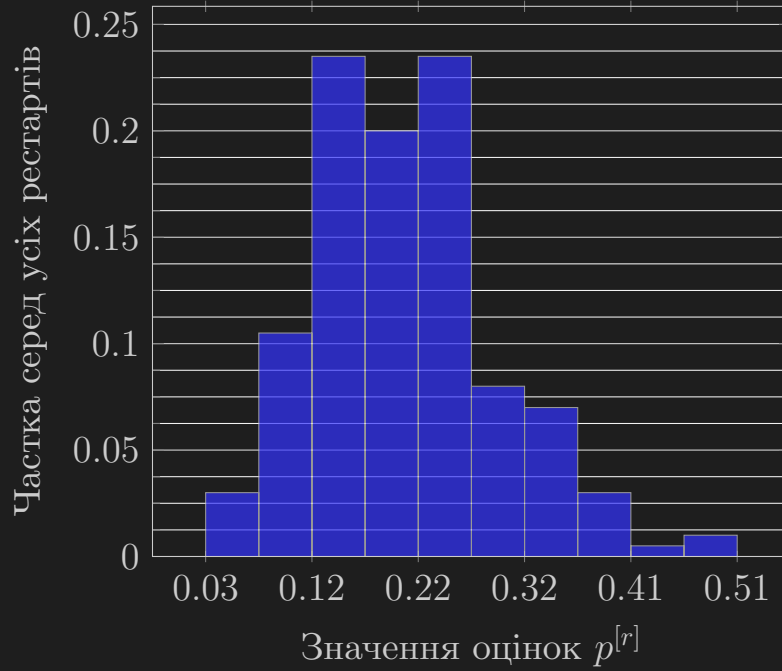
**Рисунок 3.6** – Ітерації алгоритму Баума-Велша при оцінці компонент вектора  $q$

Для досягнення точності переоцінки  $\varepsilon = 0.0001$  оцінюваного параметра  $p$  у випадку спотворених даних знадобилося  $n = 53$  ітерацій. Отримане значення  $p^{(53)} = 0.2559$  відрізняється від свого істинного значення  $p = 0.2$  на показник  $\delta = 0.0559$ . Водночас точність оцінки коефіцієнтів спотворення  $q^{(53)} = (q_1^{(53)}, q_2^{(53)}) = (0.0454, 0.1184)$  складає  $\delta = (\delta_1, \delta_2) = (0.0046, 0.0184)$ .

### 3.5.2 Висновки щодо точності оцінювання

Знову ж таки, оскільки ітераційний алгоритм Баума-Велша є збіжним лише до точки локального максимуму функції правдоподібності, виконаємо висновки щодо ефективності використаних методів для розв'язку задачі навчання шляхом аналізу результатів  $R = 200$  незалежних запусків ітераційного алгоритму.

Проілюструємо гістограму вибірки оцінок  $\{p^{[r]}\}_{r=1, \overline{R}}$  на Рис. 3.7.



**Рисунок 3.7** – Гістограма значень оцінок  $R = 200$  незалежних рестартів алгоритму Баума-Велша при спотвореності спостережень

Вибіркове середнє вказаної вибірки складає

$$\mu_p^\times = \frac{1}{R} \sum_{r=1}^R p^{[r]} = 0.2082,$$

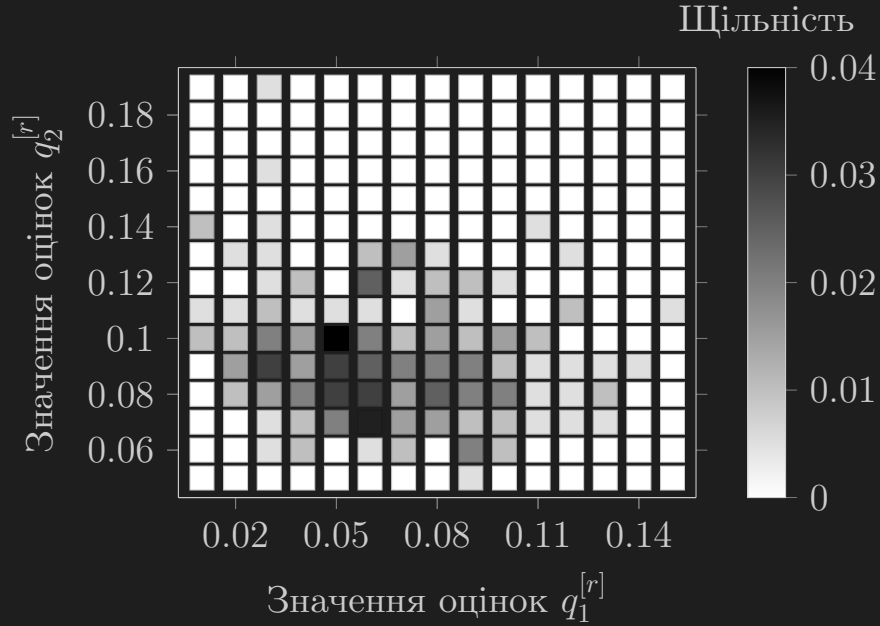
а вибіркова дисперсія

$$(S_p^\times)^2 = \frac{1}{R-1} \sum_{r=1}^R (p^{[r]} - \mu_p^\times)^2 = 0.0065$$

У порівнянні з показниками (3.3) та (3.2) помітна втрата точності ітераційного алгоритму Баума-Велша при оцінці керуючого параметра  $p$  за спотвореними спостереженнями:

$$\delta_p^\times = |p - \mu_p^\times| = |0.2 - 0.2082| = 0.0082$$

Частотна характеристика отриманих значень оцінок коефіцієнтів спотворення  $\left\{ \left( q_1^{[r]}, q_2^{[r]} \right) \right\}_{r=1, R}$  зображена на тепловій карті Рис. 3.8.



**Рисунок 3.8** – Гістограма значень оцінок ймовірностей спотворення у  $R = 200$  незалежних рестартах алгоритму Баума-Велша

Вибіркові середні складають

$$\mu_{q_1} = \frac{1}{R} \sum_{r=1}^R q_1^{[r]} = 0.0519$$

$$\mu_{q_2} = \frac{1}{R} \sum_{r=1}^R q_2^{[r]} = 0.1051$$

В той час як вибіркові дисперсії

$$S_{q_1}^2 = \frac{1}{R-1} \sum_{r=1}^R \left( q_1^{[r]} - \mu_{q_1} \right)^2 = 0.0004$$

$$S_{q_2}^2 = \frac{1}{R-1} \sum_{r=1}^R \left( q_2^{[r]} - \mu_{q_2} \right)^2 = 0.0008$$

Вказані показники свідчать про високу точність результатів:

$$\delta_{q_1} = |q_1 - \mu_{q_1}| = |0.05 - 0.0519| = 0.0019$$

$$\delta_{q_2} = |q_2 - \mu_{q_2}| = |0.1 - 0.1051| = 0.0051$$

### Висновки до розділу 3

Висновки до останнього розділу є, фактично, підсумковими під усім дослідженням; однак вони повинні стосуватись саме того, що розглядалось у розділі.

## ВИСНОВКИ

Загальні висновки до роботи повинні підсумовувати усі ваші досягнення у даному напрямку досліджень.



## ПЕРЕЛІК ПОСИЛАНЬ

1. *Koski T.* Hidden Markov models for bioinformatics. — 2002-е вид. — New York, NY : Springer, 11.2001. — (Computational Biology).
2. *Rabiner L.* A tutorial on hidden Markov models and selected applications in speech recognition // Proceedings of the IEEE. — 1989. — Т. 77, № 2. — С. 257—286. — DOI: 10.1109/5.18626. — URL: <https://doi.org/10.1109/5.18626>.
3. *Chaaraoui A. A., Climent-Pérez P., Flórez-Revuelta F.* One shot learning for gesture recognition using HMMs and hand appearance features // Pattern Recognition Letters. — 2013. — Т. 34, № 9. — С. 1009—1017.
4. *Norris J. R.* Markov Chains. — Cambridge University Press, 02.1997. — DOI: 10.1017/cbo9780511810633. — URL: <https://doi.org/10.1017/cbo9780511810633>.
5. *Nilsson M.* First Order Hidden Markov Model: Theory and Implementation Issues : тех. звіт. / Blekinge Institute of Technology, School of Engineering, Department of Signal Processing. — 2005.
6. *Larsen R. J., Marx M. L.* An introduction to mathematical statistics and its applications. — 6-е вид. — "Upper Saddle River, NJ" : "Pearson", 01.2017.
7. Finding the Jaccard Median / F. Chierichetti [та ін.] // Proceedings of the 2010 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA). — 2010. — С. 293—311. — DOI: 10.1137/1.9781611973075.25. — eprint: <https://epubs.siam.org/doi/pdf/10.1137/1.9781611973075.25>. — URL: <https://epubs.siam.org/doi/abs/10.1137/1.9781611973075.25>.

## ДОДАТОК А ГРАФІЧНИЙ ІНТЕРФЕЙС КОРИСТУВАЧА

Розробка програмного забезпечення охоплювала як імплементацію ітераційних формул переоцінки параметрів моделі, так і розробку графічного інтерфейсу користувача для ефективного керування різними блоками коду.

Наприклад, на малюнку нижче продемонстровано задання необхідних вхідних даних для генерування відповідного ланцюга Маркова для подальшого розв'язання задачі локалізації.

**Hidden Markov Model**

**Hidden chain generation**

Dimension of a state vector  $N = 5$

First initiate state  $x_0 = \text{random}$

Transition probability  $p = 0.2$

Length of a chain  $T = 200$

**VIEW STATE SHAPE**

**Observed chain generation**

Observed indices  $I = [[2,3],[1,4]]$

Implicit indices  $I^* = [1,3,5]$

Distortion coefficients  $q = \text{none}$

**VIEW OBSERVED STATE SHAPE**

**Learning algorithm**

Initial approximation  $p_0 = 0.55$

$q_0 = \text{none}$

Learning algorithm stop criterion  $20 \text{ iterations}$

Scale forward & backward coefficients  $\text{false}$

Number of restarts  $r = 1$

**RUN LEARNING ALGORITHM**

**Progress bar**

Restart №1

100%

Time of execution  $t = 4.111 \text{ s}$

**VIEW LEARNING RESULTS**

**Decoding algorithm**

Decoding algorithm mode  $\text{run for each restart}$

**RUN DECODING ALGORITHM & VIEW RESULTS**

**Estimation of implicit indexes**

Estimated implicit indexes  $I^* = (1,2,5)$

**RUN ESTIMATION & VIEW RESULTS**

Diagram showing states  $x_1, x_2, x_3, x_4, x_5$  and their connections.

Рисунок А.1 – Графічний інтерфейс користувача

Кожна з відповідних кнопок викликає блоки коду, необхідні для виконання тієї чи іншої задачі. Візуалізація результатів виконується в окремих спливаючих вікнах інтерфейсу у вигляді графіків, наведених у розділі проведення чисельного експерименту.

У лівій нижній частині інтерфейсу схематично зображена конфігурація стану ланцюга: відповідні множини спостережуваних індексів (кожна спостережувана область виокремлюється візуально за з'єднаними ребром вершинами), а також множина неявних індексів, елементи якої позначені сірим кольором.

## ДОДАТОК Б ТЕКСТИ ПРОГРАМ

У цьому додатку наведені тексти ключових інструментальних програм для проведення експериментальних досліджень. Перелік необхідних бібліотек мови Python наведений нижче:

```

1 from __future__ import annotations # enable using function specifications
2 import numpy.typing as npt
3 import typing
4
5 import numpy as np
6 import itertools
7 import copy

```

### Б.1 Обчислення коефіцієнтів прямого та зворотного ходу

```

1 def alpha_calculation(
2     y: npt.NDArray[np.int64],
3     m: npt.NDArray[np.float64],
4     A: npt.NDArray[np.float64],
5     B: npt.NDArray[np.float64],
6     T: int,
7     *args: list[float]
8 ):
9     """
10    Return forward algorithm coefficients
11
12    Parameters
13    -----
14    y : int array(T,)
15        Chain of observations
16    m : float array(pow(2,N),)
17        Initial distribution
18    A : float array(pow(2,N),pow(2,N))
19        Transition matrix
20    B : float array(pow(2,N), depends on I)
21        Emission matrix
22    T : int
23        Length of a chain
24    *args : coefficients of scaling (optional, used only for scaled forward algorithm)
25
26    Returns
27    -----
28    alpha : float array(pow(2,N),T)
29        Forward algorithm coefficients
30    P : float
31        Probability of P(Y=y)
32    scaler : float array(pow(2,N),T)
33        Coefficients of scaling (optional, used only for scaled forward algorithm)
34    """
35
36    alpha = np.zeros((T, len(B)))

```

```

37     if len(args) == 0:
38         for t in range(T):
39             for i in range(len(B)):
40                 if t == 0:
41                     alpha[t][i] = m[i]*B[i][y[t]]
42                 else:
43                     aA = 0.0
44                     for j in range(len(B)):
45                         aA += alpha[t-1][j]*A[j][i]
46                     alpha[t][i] = aA*B[i][y[t]]
47
48     P = 0
49     for i in range(len(alpha[T-1])):
50         P += alpha[T-1][i]
51
52     return alpha, P
53
54 if len(args) != 0:
55     for t in range(T):
56         for i in range(len(B)):
57             if t == 0:
58                 alpha[t][i] = m[i]*B[i][y[t]]
59                 args[0][t] += alpha[t][i]
60             else:
61                 aA = 0.0
62                 for j in range(len(B)):
63                     aA += alpha[t-1][j]*A[j][i]
64                 alpha[t][i] = aA*B[i][y[t]]
65                 args[0][t] += alpha[t][i]
66
67         for i in range(len(B)):
68             alpha[t][i] = alpha[t][i]/args[0][t]
69
70     P = 0
71     for t in range(T):
72         P += np.log(args[0][t])
73
74     return alpha, P, args[0]

```

```

1 def beta_calculation(
2     y: npt.NDArray[np.int64],
3     A: npt.NDArray[np.float64],
4     B: npt.NDArray[np.float64],
5     T: int,
6     *args: list[float]
7 ):
8     """
9     Return backward algorithm coefficients
10
11     Parameters
12     -----
13     y      : int array(T,)
14              Chain of observations
15     A      : float array(pow(2,N),pow(2,N))
16              Transition matrix
17     B      : float array(pow(2,N), depends on I)
18              Emission matrix
19     T      : int
20              Length of a chain
21     *args  : coefficients of scaling (optional, used only for scaled backward algorithm)
22

```

```

23 Returns
24 -----
25 beta : float array(pow(2,N),T)
26         Backward algorithm coefficients
27 """
28
29 beta = np.zeros((T, len(B)))
30
31 if len(args) == 0:
32     for t in range(T-1, -1, -1):
33         for i in range(len(B)):
34             if t == T-1:
35                 beta[t][i] = 1
36             else:
37                 bAB = 0.0
38                 for j in range(len(B)):
39                     bAB += beta[t+1][j]*A[i][j]*B[j][y[t+1]]
40                 beta[t][i] = bAB
41
42 if len(args) != 0:
43     for t in range(T-1, -1, -1):
44         for i in range(len(B)):
45             if t == T-1:
46                 beta[t][i] = 1
47             else:
48                 bAB = 0.0
49                 for j in range(len(B)):
50                     bAB += beta[t+1][j]*A[i][j]*B[j][y[t+1]]
51                 beta[t][i] = bAB
52
53     for i in range(len(B)):
54         beta[t][i] = beta[t][i]/args[0][t]
55
56 return beta

```

## Б.2 Алгоритм Баума-Велша

```

1 def learning_algorithm(
2     y: npt.NDArray[np.int64],
3     N: int,
4     T: int,
5     I: list[list[int]],
6     estimator: typing.Literal[
7         "parameter p estimation task (distortion-free model)",
8         "parameter p estimation task (model with distortion)",
9         "parameter p and coefficients q estimation task"
10    ],
11     p0: float,
12     q0: float,
13     scaling: bool
14 ):
15     """
16     Return learning algorithm results (estimated parameters)
17
18     Parameters
19     -----
20     y : int array(T,)
21         Chain of observations
22

```

```

23     N : int
24         Dimention of any state vector
25     T : int
26         Length of a chain
27     I : int array
28         Set of observed indices
29     estimator : string
30         Type of estimation (only p or both p & q estimation)
31     p0 : float
32         Initial approximation of parameter p
33     q0 : float array(len(I),)
34         Initial approximation of distortion coefficients q
35     scaling : string
36         Either to scale forward and backward coefficients or not
37
38     Returns
39     -----
40     parameter : float array
41         List of estimated parameters [p,q] for each iteration
42     joint_probabilities : float array
43         List of P(Y=y) probabilities (concerns to forward algorithm)
44     joint_probabilities_increments : float array
45         List of P(Y=y) increments through iterations (concerns to forward algorithm)
46     """
47
48     joint_probabilities = []
49     joint_probabilities_increments = []
50
51     if estimator == "parameter p and coefficients q estimation task":
52         parameter = []
53         parameter.append([p0, q0])
54
55         p = copy.deepcopy(parameter[0][0])
56         q = copy.deepcopy(parameter[0][1])
57     else:
58         parameter = []
59         parameter.append([p0])
60
61         p = copy.deepcopy(parameter[0][0])
62
63     number_of_iterations = 0
64
65     while (
66         number_of_iterations < 20 or
67         abs(1-joint_probabilities[-1]/joint_probabilities[-2]) > 0.0001
68     ):
69         if estimator == "parameter p estimation task (distortion-free model)":
70             m,A,B = probability_measures_of_HMM(p,N,I)
71         elif estimator == "parameter p estimation task (model with distortion)":
72             m,A,B = probability_measures_of_HMM(p,N,I,q0)
73         elif estimator == "parameter p and coefficients q estimation task":
74             m,A,B = probability_measures_of_HMM(p,N,I,q)
75
76         if scaling == "false":
77             alpha,P = alpha_calculation(y,m,A,B,T)
78             beta = beta_calculation(y,A,B,T)
79         else:
80             scaler = np.zeros(T)
81             alpha,P,scaler = alpha_calculation(y,m,A,B,T,scaler)
82             beta = beta_calculation(y,A,B,T,scaler)
83
84         joint_probabilities.append(copy.deepcopy(P))
85
86         if number_of_iterations == 0:
87             joint_probabilities_increments.append(copy.deepcopy(P))

```

```

88     else:
89         joint_probabilities_increments.append(
90             abs(copy.deepcopy(P) - joint_probabilities_increments[-1])
91         )
92
93     p_numerator = calculate_p_numerator(y,alpha,beta,A,B,N,T)
94     p_denominator = calculate_p_denominator(alpha,N,T)
95     p = p_numerator/p_denominator
96
97     parameter.append([copy.deepcopy(p)])
98
99     if estimator == "parameter p and coefficients q estimation task":
100         for j in range(len(I)):
101             part_1 = calculate_q_estimation_part_1(y,q[j],alpha,beta,A,N,T,I,j)
102             part_2 = calculate_q_estimation_part_2(y,q[j],alpha,beta,A,N,T,I,j)
103
104             qj_numerator = copy.deepcopy(part_1)
105             qj_denominator = copy.deepcopy(part_1) + copy.deepcopy(part_2)
106
107             q[j] = qj_numerator/qj_denominator
108
109             parameter[number_of_iterations + 1].append(copy.deepcopy(q))
110
111         number_of_iterations += 1
112
113     return parameter, joint_probabilities, joint_probabilities_increments

```

## Б.3 Алгоритм Вітербі

```

1 def viterbi(
2     y: npt.NDArray[np.int64],
3     m: npt.NDArray[np.float64],
4     A: npt.NDArray[np.float64],
5     B: npt.NDArray[np.float64],
6     T: int
7 ) -> npt.NDArray[np.int64]:
8     """
9     Return decoded state chain
10
11     Parameters
12     -----
13     y : int array(T,)
14         Chain of observations
15     m : float array(pow(2,N),)
16         Initial distribution
17     A : float array(pow(2,N),pow(2,N))
18         Transition matrix
19     B : float array(pow(2,N), depends on I)
20         Emission matrix
21     T : int
22         Length of a chain
23     *args : coefficients of scaling (optional, used only for scaled forward algorithm)
24
25     Returns
26     -----
27     x : int array(T,)
28         Decoded enumerated state chain
29     """
30

```



```

31 delta = [[0.0 for i in range(len(B))] for t in range(T)]
32 psi = [[0 for i in range(len(B))] for t in range(T)]
33
34 for t in range(T):
35     for i in range(len(B)):
36         if t == 0:
37             delta[t][i] = m[i]*B[i][y[t]]
38         else:
39             dA = []
40             for j in range(len(B)):
41                 dA.append(delta[t-1][j]*A[j][i]*B[i][y[t]])
42             delta[t][i] = max(dA)
43             psi[t][i] = np.argmax(dA)
44
45 x = []
46 delta_hat = max(delta[T-1])
47 x.append(np.argmax(delta[T-1]))
48
49 for t in range(T-2, -1, -1):
50     x.insert(0, psi[t+1][x[0]])
51
52 return x

```

## Б.4 Алгоритм розв'язку задачі локалізації

```

1 def define_distance(
2     A: npt.NDArray[np.float64],
3     B: npt.NDArray[np.float64],
4     label: typing.Literal["square", "weighted Jaccard"]
5 ):
6     """
7     Return a distance between set A and set B
8     """
9
10    if label == "square":
11        return sum([pow(a-b,2) for a,b in zip(A,B)])
12    if label == "weighted Jaccard":
13        return 1 - sum([min(a,b) for a,b in zip(A,B)])/sum([max(a,b) for a,b in zip(A,B)])
14
15 def estimate_implicit_indices(
16     x_real: list[str],
17     x_predicted: list[int],
18     real_implicit_indices: list[int],
19     estimate_length: typing.Literal["maximum", "onsistent"],
20     T: int,
21     N: int,
22     metrics_type: typing.Literal["square", "weighted Jaccard"]
23 ) -> tuple[list]:
24     """
25     Return estimation of implicit indices in a hidden chain
26
27     Parameters
28     -----
29     x_real : string array(T,)
30         Real chain of the hidden states
31     x_predicted : int array(T,)
32         Decoded enumerated state chain
33     real_implicit_indices : int array
34         Real set of implicit indices

```

```

35 estimate_length : string array()
36     A way to estimate length of set of implicit indices:
37     either by "maximum" method or by using "consistent" estimation
38 T : int
39     Length of a chain
40 N : int
41     Dimention of any state vector
42 metrics_type : string
43     Either "square" or "weighted Jaccard"
44
45 Returns
46 -----
47 predicted_implicit_indices : int array(T,)
48     Set of predicted_implicit_indices
49     (considering an acceptable list of "0", "1", "2" etc. for given combination)
50 """
51
52 phi_real = np.zeros(T, dtype=int)
53 for t in range(T):
54     phi_real[t] = sum([int(list(x_real[t])[i]) for i in real_implicit_indices])
55
56 if estimate_length[0] == "maximum":
57     estimated_length = max(phi_real)
58 elif estimate_length[0] == "consistent":
59     p = estimate_length[1]
60     estimated_length = int(
61         (N/(1-p))*(1 - sum([1 for t in range(T-1) if phi_real[t] == phi_real[t+1]])/(T-1))
62     )
63
64 offered_implicit_indices = list(itertools.combinations([i for i in range(N)], estimated_length))
65
66 metric = np.zeros(len(offered_implicit_indices))
67 for k in range(len(offered_implicit_indices)):
68     phi_offered = np.zeros(T, dtype=int)
69     for t in range(T):
70         phi_offered[t] = sum([int(list(x_predicted[t])[i]) for i in offered_implicit_indices[k]])
71
72     metric[k] = define_distance(phi_real, phi_offered, metrics_type)
73
74 min_metric_value = min(metric)
75 argmin_metric_value = [
76     index for index in range(len(metric)) if metric[index] == min_metric_value
77 ]
78
79 predicted_implicit_indices = []
80 for index in argmin_metric_value:
81     predicted_implicit_indices.append(offered_implicit_indices[index])
82
83 return predicted_implicit_indices[0]

```