УДК 519.816

Михалевич В. М., Янівський О. М.

ДО МЕТОДУ АПРОКСИМАЦІЇ НЕЧІТКОЇ СТАТИСТИЧНОЇ ЗАКОНОМІРНОСТІ

Розроблено один із підходів до «статистичного означення» статистичної закономірності та методу апроксимації статистичної закономірності. Запропоновано поняття нечіткої статистичної закономірності, що узагальнює введене раніше поняття статистичної закономірності [1]. Запропоновано програмну реалізацію знаходження статистичної закономірності послідовності елементів скінченного алфавіту. Апарат статистичних закономірностей використовується при побудові загальної теорії рішень, що дає змогу розглядати задачі, які виходять за межі класичної теорії статистичних рішень.

Ключові слова: випадковість, невизначеність, масове явище, статистична закономірність.

Вступ

До останнього часу, окрім абстрактного означення статистичної закономірності (в теорії ймовірностей — аналог аксіоматики Колмогорова), було відсутнє емпіричне визначення статистичної закономірності (в теорії ймовірностей — статистичне означення теорії ймовірності на підставі Закону великих чисел, як границі відносної частоти відповідної події) і програмна реалізація методів статистичного оцінювання.

Сукупності, які досліджуються статистикою, охоплюють факти, яким притаманні індивідуальні випадкові особливості. У величезній кількості спостережень і зіставлень такі особливості, які відносяться тільки до окремих фактів і не характерні для сутності всієї досліджуваної маси, взаємно погашаються і таким чином проявляється статистична закономірність у всій її якісно-кількісній визначеності як результат основних і істотних причин.

Необхідність і випадковість не тільки нерозривно пов'язані між собою, вони є єдністю протилежностей, які переходять одна в одну, причому випадковість — це форма прояву необхідності. Основну закономірність у кожному окремому факті можуть спотворити або затушувати дії випадкових обставин, але в досить великій кількості подій (масовість) необхідність буде виявлена повною мірою.

Далеко не завжди, як і в стохастичному випадку, вдається точно обрахувати статистичну закономірність. Тому при стохастичній невизначеності переходять до статистичного оцінювання характеристик розподілу, тобто до наближеного (залежно від розмірності вибірки) аналізу масового явища. І це предмет математичної статистики. А при невизначеності в широкому сенсі (тобто в масових явищах, у яких ми цікавимося лише їх статистичними закономірностями) для оцінки статистичної закономірності послідовності елементів скінченного алфавіту (простір елементарних подій) необхідно було запропонувати алгоритм, що, по заданій цією послідовністю вибірці певного обсягу, формує наближення цієї статистичної закономірності із заданою точністю.

При цьому послідовність можна задавати або формально (аналітично), або маючи процедуру обчислення її загального елемента, або конкретно здійснюючи потенційну можливість послідовного (із довільною кількістю повторів) експерименту з невизначеністю результатів у широкому сенсі.

Запропонований алгоритм наближення статистичної закономірності заданої послідовності полягає в тому, що по отриманій послідовній вибірці елементів скінченного алфавіту цієї послідовності, які для спрощення вважатимемо нулем або одиницею, розмірності $2n$ знаходимо послідовності відносних частот появ нуля (відповідно одиниці). Аналізуємо цю послідовність, починаючи з $n+1$ елемента, на предмет знаходження граничних точок.

Тематика цієї роботи нова та орієнтована на розвиток та практичне застосування математичної статистики статистичних закономірностей у задачах прийняття рішень, зокрема на основі принципу гарантованого результату [2; 3].

Модель з фрактальним активним часом: оцінка справедливої ціни опціонів та управління ризиком

Нагадаємо означення статистичної закономірності [1]. Нехай X — довільна множина, $f : X \to R$ — обмежена дійснозначна функція, $\bar{x} \in X^N$ — довільна послідовність. Поставимо у відповідність будь-якому натуральному n та будь-якій підмножині $A \in 2^X$ частоту потрапляння в A перших n членів послідовності $\bar{x} = (\bar{x}_1, \bar{x}_2, \ldots, \bar{x}_n, \ldots)$

$$p_{\bar{x}}^{(n)}(A) = \frac{1}{n} \sum_{i=1}^{n} I_a(\bar{x}_i). \qquad (1)$$

Отримаємо послідовність $\{p_{\bar{x}}^{(n)}; n \in N\}$ імовірнісних розподілів на 2^X, яка завжди має непорожню множину граничних точок — скінчено-адитивних імовірнісних розподілів на $\mathcal{A}$, що представляє статистичну закономірність послідовності $\bar{x}$.

Очевидно, будь-якій послідовності $\bar{x} \in X^N$ можна співставити вибіркову спрямованість $(\varphi|N, \geq)$ в X, де $(\geq)$ — природний напрям незростання в N. А саме $\varphi(n) = (\bar{x}_1, \bar{x}_2, \ldots, \bar{x}_n)$. При цьому $p_\varphi(n) = p_{\bar{x}}^{(n)}$, $n \in N$, де

$$p_{\bar{x}}^{(n)}(A) = \frac{1}{n} \sum_{i=1}^{n} I_a(\bar{x}_i), \quad \forall A \in \mathcal{A}. \qquad (2)$$

Отже, $P_\varphi = P_{\bar{x}}$.

При цьому клас усіх статистичних закономірностей на множині X значно ширший від класу всіх статистичних закономірностей послідовностей елементів множини X, бо закономірністю послідовності елементів множини X може бути тільки закономірність, зосереджена на зліченній підмножині множини X, тобто для кожного $p \in P_{\bar{x}}$ справедливе співвідношення $p(X_0) = 1$, де $X_0 \subseteq X$ становить множину значень послідовності $\bar{x}$.

Алгоритм приймає на вході список відносних частот та точність, з якою потрібно провести обчислення, а на виході повертає список пар, що складаються з проміжного значення частоти та числа, що характеризує кількість частот, що лежали в одних межах точності.

Роботу алгоритму можна уявити таким чином:

1. Потрібно відкинути першу половину значень відносних частот, отриманих на вході.
2. Потрібно створити проміжок $[0, 1]$, який розділено на підпроміжки, довжина яких дорівнює точності, яку було передано, як аргумент. Припустимо, що точність дорівнює a, тоді відбудеться таке розбиття: $[0, a]$, $[a, 2a]$, …, $[ak, 1]$.
3. У кожного підпроміжку є свій лічильник, який вказує на те, скільки частот було в заданих межах точності.
4. У кожного підпроміжку є змінна для зберігання значення відносної частоти.
5. Кожне нове значення відносної частоти потрібно віднести до конкретного проміжку зі списку проміжків, який було описано вище.
6. Кожного разу перезаписується значення відносної частоти, тобто на виході проміжок міститиме інформацію лише про останнє значення відносної частоти, що зберігалося в заданому проміжку.

Відомості щодо реалізації:

1. Дані зберігатимуться в списку пар (значення, кількість). Цей список — спрощене представлення набору підпроміжків, який можна отримати після розбиття згідно із заданою точністю. За допомогою позиції в списку можна визначити точні межі заданого проміжку. Довжина списку рівна кількості проміжків, на який розбивається проміжок $[0, 1]$.
2. Далі слід виконати ітерацію по списку значень відносних частот. Для кожного елемента списку слід обрахувати номер проміжку, до якого він відноситься. Наступний крок — оновлення даних у списку: збільшення кількості та перезапис значення.

Відповідно для програмної реалізації алгоритму потрібно реалізувати виконання таких завдань:

1. визначити кількість проміжків, на які слід розбити проміжок $[0, 1]$: потрібно поділити одиницю на задану точність та повернути найменше ціле число, що є більшим або рівним результату ділення;
2. визначити номер проміжку, до якого слід віднести певне значення відносної частоти: слід поділити задане значення ймовірності на задану точність та відкинути дробову частину.

Кроки алгоритму:

1. Відкинути половину значень відносних частот, отриманих на вході.
2. Обрахувати довжину списку на основі заданої точності.
3. Виконати для кожного елемента списку значень відносних частот такі кроки:

a) обрахувати номер проміжку, до якого відноситься значення відносної частоти;

b) збільшити значення кількості елементів, що потрапляють у заданий проміжок;

c) оновити значення відносної частоти для заданого проміжку.

4. Створити новий список на основі попереднього списку, в якому не буде проміжків, що не містять елементів.

```java
public static List<PointHolder>
getPoints(List<Double> sequence, double epselon) {
  int numberOfBuckets = (int) Math.ceil(1 / epselon);

  Double[] values = new Double[numberOfBuckets];
  int[] counts = new int[numberOfBuckets];

  List<PointHolder> result = new ArrayList<>();
  for (int i = sequence.size() / 2; i < sequence.size(); i++) {
    int index = (int) (sequence.get(i) / epselon);
    values[index] = sequence.get(i);
    ++counts[index];
  }

  for (int i = 0; i < values.length; i++)
    if (values[i] != null)
      result.add(new PointHolder(counts[i], values[i]));

  return result;
}
```

Рис. 1. Приклад реалізації описаного алгоритму

Як приклад розглянемо послідовність, статистична закономірність якої була отримана в роботі [1]. Дано таку послідовність елементів множини $\Omega = \{0, 1\}$:

$$\bar{\theta}_0 := 01010011000011110000\ldots \qquad (3)$$

Принцип побудови послідовності (3) такий: ця послідовність складається із серії нулів та одиниць, причому серії одиниць слідують за серіями нулів і кожна серія одиниць має довжину серії нулів, що стоїть перед нею; перша серія нулів складається з одного члена, а кожна наступна серія нулів містить стільки нулів, скільки всі попередні серії разом узяті.

Статистична закономірність $P(\bar{\theta}_0)$ даної послідовності має такий вигляд:

$$\left\{ p : p\big(\{0\}\big) = \alpha, \alpha \in \left[\frac{1}{2}, \frac{2}{3}\right] \right\}. \qquad (4)$$

Аналітичне представлення цієї послідовності можна задати в такому вигляді:

$$\mathrm{greater}\Big(\{i\} - 2^{\mathrm{floor}\left(\frac{\ln(\{i\})}{\ln(2)}\right)}, 2^{\mathrm{floor}\left(\frac{\ln(\{i\})}{\ln(2)}\right) \operatorname{div} 2}\Big), \quad (5)$$

де i — номер елемента в послідовності, floor — функція, що повертає цілу частину від заданого числа, $\ln$ — функція, що обраховує натуральний логарифм, $\mathrm{greater}$ — функція, що повертає одиницю, якщо перший аргумент більший за другий, в іншому випадку — нуль, div — цілочисельне ділення.

Використовуючи аналітичне представлення (5), можна згенерувати послідовність (3). Наступний крок — створення послідовності частот, яку можна передати на вхід запропонованому алгоритму. Для цього потрібно зберігати інформацію на кожному кроці про загальну кількість елементів у послідовності і кількість елементів, що рівні заданому на вході, поділити друге значення на перше та записати в результуючий список. Надалі для обрахунків буде використано точність 0,01.

```java
public static List<Double>
generateProbabilitySequence(List<Double> sequence, double value) {
  List<Double> result = new ArrayList<>();

  int quantity = 0;
  int total = 0;

  for (double i : sequence) {
    if (i == value)
      ++quantity;
    ++total;
    result.add((double) quantity / total);
  }

  return result;
}
```

Рис. 2. Приклад програмної реалізації створення послідовності частот (на мові програмування Java)

Таблиця 1. Приклад результату роботи алгоритму

0,5099892331618615
0,5199812492675495
0,5299518016984164
0,5399820305480683
0,5499890134036476
0,5599957030830379
0,5699737532808399
0,5799835931091059
0,58998998998999
0,5904
0,6000585994726048
0,6100685135537682
0,6200423857099606
0,6300569143208737
0,64
0,6500555467386129
0,6600064453754432

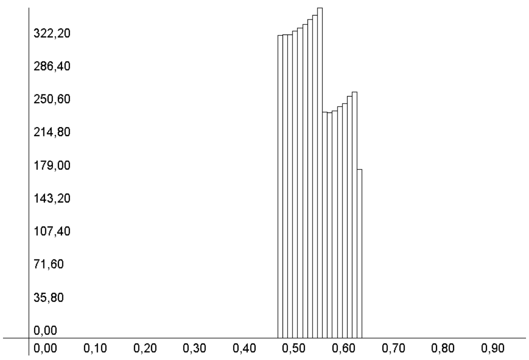

Рис. 3. Відображення результатів роботи алгоритму

Застосовуючи алгоритм до послідовності частот, отриманої за допомогою формули (5), отримано такий результат, який зображено в таблиці 1. Порівнюючи результат роботи алгоритму з реальною статистичною закономірністю, ми бачимо повний збіг при збільшенні точності і обсягу вибірки. На рис. 3 подано графічне відображення результатів роботи алгоритму, де вісь абсцис — це значення відносної частоти, а вісь ординат — це кількість елементів, що перебувають в одних межах. Цю кількість елементів можна застосувати як значення функції достовірності при означенні нечіткої статистичної закономірності.

Список літератури

1. Михалевич В. М. До побудови закономірності статистично нестійкої послідовності елементів скінченної множини / В. М. Михалевич // Наукові записки НаУКМА. — 2014. — Т. 152 : Фізико-математичні науки. — С. 38–40.
2. Михалевич В. М. О двух критериях при поэтапном выборе решений / В. М. Михалевич // Кибернетика и системный анализ. — 2012. — № 2. — С. 68–79.
3. Михалевич В. М. Оптимальность, сочетающая принципы гарантированного и наилучшего результата, в задачах принятия решений / В. М. Михалевич // Теорія оптимальних рішень. — 2012. — № 11. — С. 53–59.

V. Mykhalevych, O. Yanivskyy

TO THE METHOD OF THE FUZZY STATISTICAL CONFORMITY APPROXIMATION

One of the approaches to the "statistical definition" of the statistical conformity and the method of the statistical conformity approximation is developed. The notion of fuzzy statistical conformity that generalizes the stated above notion of statistical conformity is proposed. The program implementation finding of the finite alphabet elements sequence statistical conformity is proposed as well. The statistical conformity instrument is used for general decision-making theory construction that provides the opportunity to consider problems that go beyond the limits of the classic statistical decisions theory.

Keywords: probability, statistical uncertainty, mass phenomenon, statistical conformity.

Матеріал надійшов 13.09.2017