И.В. СЕРГИЕНКО, О.А. ПРОВОТАР

УДК 681.3

РАСПОЗНАВАНИЕ ВТОРИЧНОЙ СТРУКТУРЫ ДНК НЕЧЕТКИМИ СИСТЕМАМИ ЛОГИЧЕСКОГО ВЫВОДА

Аннотация. Рассмотрены вопросы построения нечетких систем логического вывода для распознавания вторичной структуры ДНК. Приведен пример предвидения структуры центрального остатка белка MutS как выхода нечеткой системы с процедурой логического вывода Мамдани.

Ключевые слова: нечеткая система, вторичная структура ДНК, нечеткое множество.

ВВЕДЕНИЕ

Известно [1, 2], что пространственная структура ДНК определяется ее аминокислотной последовательностью. В свою очередь, пространственная структура определяет функциональность белка.

Задача распознавания структур белка различных уровней организации является достаточно сложной. Для ее решения используются различные методы и подходы, в том числе экспериментальные (основанные на физике образования химических связей), машинного обучения (используются базы данных экспериментально найденных вторичных структур как обучающих выборок), вероятностные (на основе байесовских процедур и цепей Маркова).

В настоящей статье предлагается метод распознавания вторичной структуры ДНК с помощью нечетких систем логического вывода. Задача состоит в следующем: необходимо построить нечеткую систему логического вывода, которая по произвольной аминокислотной последовательности определяла бы (в виде нечеткого множества) вторичную структуру центрального остатка (аминокислоты) входной последовательности.

Для решения этой задачи, в первую очередь, необходимо спроектировать нечеткую систему по обучающим выборкам.

ПОСТРОЕНИЕ НЕЧЕТКИХ ПРАВИЛ

Один из методов построения системы нечетких правил по числовым данным приведен в [3]. Он заключается в следующем. Пусть для простоты создается база правил с двумя входами и одним выходом. Для этого необходимы обучающие данные (выборки) в виде множества

$$(x_1(i), x_2(i); d(i)), \quad i = 1, 2, \ldots, m,$$

где $x_1(i)$, $x_2(i)$ — входы модуля нечеткого управления, $d(i)$ — выход модуля нечеткого управления. Задача состоит в построении таких нечетких правил, чтобы образованная на их основе нечеткая система логического вывода по входным данным генерировала корректные выходные данные. Алгоритм решения сформулированной задачи сводится к следующей последовательности шагов.

Шаг 1. Разделение пространства входов и выходов на области и определение соответствующих функций принадлежности. Каждый вход и выход делим на $2N+1$ отрезок, где N — число, которое для каждого входа подбирается индивидуально. Отдельные области (отрезки) будем обозначать следующим образом:

$$M_N \text{ (левая } N), \dots, M_1 \text{(левая 1)}, S \text{(средняя)}, D_1 \text{(правая 1)}, \dots, D_N \text{ (правая } N).$$

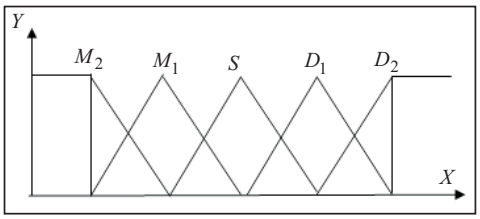

Рис. 1

Каждая функция принадлежности имеет определенную форму, например треугольную (рис. 1).

Шаг 2. Построение нечетких правил на основе обучающих выборок. Определяются меры принадлежности обучающих данных $(x_1(i), x_2(i); d(i))$ в каждой области. Для каждого набора обучающих данных нечеткое правило логического вывода формируется следующим образом:

$$(x_1(i), x_2(i); d(i)) \Rightarrow (x_1(i)[\max \text{ в соответствующей области } S_1],$$

$$x_2(i)[\max \text{ в соответствующей области } S_2],$$

$$d(i)[\max \text{ в соответствующей области } S_3]) \Rightarrow$$

$$\Rightarrow R^{(i)}: \textbf{ if } x_1 \text{ есть } S_1 \textbf{ and } x_2 \text{ есть } S_2 \textbf{ then } y \text{ есть } S_3.$$

Шаг 3. Элиминация противоречий. Некоторые правила могут быть противоречивыми. Один из способов решения этой проблемы заключается в приписывании каждому правилу степени истинности с последующим выбором правила с наибольшей мерой принадлежности. Например, для правила вида

$$R: \textbf{ if } x_1 \text{ есть } A_1 \textbf{ and } x_2 \text{ есть } A_2 \textbf{ then } y \text{ есть } B$$

мера истинности определяется как

$$SR(R) = \mu_{A_1}(x_1)\,\mu_{A_2}(x_2)\,\mu_B(y).$$

Шаг 4. Дефаззификация. Происходит при определении количественных значений выходной величины.

МОДИФИЦИРОВАННЫЙ АЛГОРИТМ ПОСТРОЕНИЯ НЕЧЕТКИХ ПРАВИЛ

Пусть создается база правил с n входами и одним выходом с использованием имеющихся обучающих данных (выборок) в виде множества пар

$$(x_1(i), x_2(i), \dots, x_n(i);\ d(i)),\ \ i = 1, 2, \dots, m,$$

где $x_j(i)$ — входы модуля нечеткого управления, $x_j(i) \in \{a_1, a_2, \dots, a_k\}$; $d(i)$ — выход модуля нечеткого управления, $d(i) \in \{b_1, b_2, \dots, b_l\}$. Следует построить на основе обучающих данных нечеткую систему логического вывода, которая по произвольным входным данным генерировала бы корректные исходные данные.

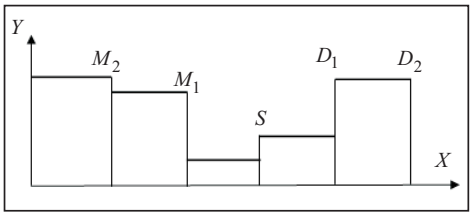

Рис. 2

Шаг 1. Разделение пространства входов и выходов на области. Разделяем обучающие данные на группы по $m_1, \dots, m_k$ строк, т.е. каждый вход и выход делим на $2N+1$ отрезок, где N подбирается индивидуально для каждого входа.

Отдельные области (отрезки) будем обозначать следующим образом:

$$M_N \text{ (левая } N), \dots, M_1 \text{(левая 1)}, S \text{(средняя)}, D_1 \text{(правая 1)}, \dots, D_N \text{ (правая } N).$$

Определим для каждой области функцию принадлежности. В случае $N = 2$ функция принадлежности может иметь вид, представленный на рис. 2.

Шаг 2. Построение нечетких множеств на основе обучающих выборок. Каждой группе m_i обучающих данных

$$(x_1(1), x_2(1), \dots, x_n(1); d(1)),$$
$$(x_1(2), x_2(2), \dots, x_n(2); d(2)),$$
$$(x_1(m_i), x_2(m_i), \dots, x_n(m_i); d(m_i))$$

сопоставляем нечеткие множества вида

$$A_1^{m_i} = \frac{|a_1^{(1)}|}{m_i} / a_1 + \dots + \frac{|a_k^{(1)}|}{m_i} / a_k,$$
$$\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots$$
$$A_n^{m_i} = \frac{|a_1^{(n)}|}{m_i} / a_1 + \dots + \frac{|a_k^{(n)}|}{m_i} / a_k,$$
$$B^{m_i} = \frac{|b_1|}{m_i} / b_1 + \dots + \frac{|b_l|}{m_i} / b_l,$$

где $|a_1^{(j)}|$ — количество символов a_1 в j-м столбце группы обучающих данных, $|b_j|$ — количество символов b_j в последнем столбце группы обучающих данных.

Шаг 3. Построение нечетких правил на основе нечетких множеств. Осуществляется на основании предыдущего шага по следующей схеме:

$$\left.\begin{matrix}(x_1(1), x_2(1), \dots, x_n(1); d(1))\\ (x_1(2), x_2(2), \dots, x_n(2); d(2))\\ \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\\ (x_1(m_1), x_2(m_1), \dots, x_n(m_1); d(m_1))\end{matrix}\right\} \Rightarrow$$

$$\Rightarrow \boldsymbol{R^{(1)}}: \textbf{ if } x_1 \text{ есть } A_1^{m_1} \textbf{ and } x_2 \text{ есть } A_2^{m_1} \textbf{ and } \dots \textbf{ and } x_n \text{ есть } A_n^{m_1} \textbf{ then } y \text{ есть } B^{m_i}.$$

Шаг 4. Элиминация противоречий.

Шаг 5. Дефаззификация. Проводится при определении количественных значений выходной величины.

Вышеприведенный алгоритм каждому набору обучающих данных ставит в соответствие нечеткое правило логического вывода.

Пример. Покажем, каким образом использовать предложенный алгоритм построения нечетких правил для распознавания вторичной структуры ДНК.

Известно [1, 2], что вторичная структура фрагментов полипептидной последовательности определяется в основном взаимодействиями соседних аминокислот в пределах этих фрагментов. Точнее говоря, тип вторичной структуры конкретного остатка определяется исходя из окружения соседних аминокислот.

Для построения нечетких правил логического вывода используются обучающие выборки, состоящие из 15 остатков белка MutS [2], которые имеют следующую последовательность:

```
K V S E G G L I R E G Y D P D
e - - - h h h h h h h h h h h

V S E G G L I R E G Y D P D L
- - - h h h h h h h h h h h h

S E G G L I R E G Y D P D L D
- - h h h h h h h h h h h h h

E G G L I R E G Y D P D L D A
- h h h h h h h h h h h h h h

G G L I R E G Y D P D L D A L
h h h h h h h h h h h h h h h
```

Распознавание относится к центральному остатку, вторичная структура *h* используется для спирали, *e* — для цилиндра, символ "–" применяется для обозначения иной структуры.

Согласно алгоритму разделяем обучающие данные, например на три группы:

```
K  V  S  E  G  G  L  h  R  E  G  Y  D  P  D
V  S  E  G  G  L  I  h  E  G  Y  D  P  D  L;
S  E  G  G  L  I  R  h  G  Y  D  P  D  L  D
E  G  G  L  I  R  E  h  Y  D  P  D  L  D  A;
G  G  L  I  R  E  G  h  D  P  D  L  D  A  L.
```

Сопоставим каждой группе соответствующие нечеткие множества:

1-я группа:

$$A_1^{(m_1)} = \frac{1}{2}/K + \frac{1}{2}/V, \quad A_6^{(m_1)} = \frac{1}{2}/G + \frac{1}{2}/L, \quad A_{11}^{(m_1)} = \frac{1}{2}/Y + \frac{1}{2}/D,$$
$$A_2^{(m_1)} = \frac{1}{2}/V + \frac{1}{2}/S, \quad A_7^{(m_1)} = \frac{1}{2}/L + \frac{1}{2}/I, \quad A_{12}^{(m_1)} = \frac{1}{2}/D + \frac{1}{2}/P,$$
$$A_3^{(m_1)} = \frac{1}{2}/S + \frac{1}{2}/E, \quad A_8^{(m_1)} = \frac{1}{2}/R + \frac{1}{2}/E, \quad A_{13}^{(m_1)} = \frac{1}{2}/P + \frac{1}{2}/D,$$
$$A_4^{(m_1)} = \frac{1}{2}/E + \frac{1}{2}/G, \quad A_9^{(m_1)} = \frac{1}{2}/E + \frac{1}{2}/G, \quad A_{14}^{(m_1)} = \frac{1}{2}/D + \frac{1}{2}/L.$$
$$A_5^{(m_1)} = 1/G, \quad A_{10}^{(m_1)} = \frac{1}{2}/G + \frac{1}{2}/Y,$$

2-я группа

$$A_1^{(m_2)} = \frac{1}{2}/S + \frac{1}{2}/E, \quad A_6^{(m_2)} = \frac{1}{2}/I + \frac{1}{2}/R, \quad A_{11}^{(m_2)} = \frac{1}{2}/P + \frac{1}{2}/D,$$
$$A_2^{(m_2)} = \frac{1}{2}/E + \frac{1}{2}/G, \quad A_7^{(m_2)} = \frac{1}{2}/R + \frac{1}{2}/E, \quad A_{12}^{(m_2)} = \frac{1}{2}/D + \frac{1}{2}/L,$$
$$A_3^{(m_2)} = 1/G, \quad A_8^{(m_2)} = \frac{1}{2}/G + \frac{1}{2}/Y, \quad A_{13}^{(m_2)} = \frac{1}{2}/L + \frac{1}{2}/D,$$
$$A_4^{(m_2)} = \frac{1}{2}/G + \frac{1}{2}/L, \quad A_9^{(m_2)} = \frac{1}{2}/Y + \frac{1}{2}/D, \quad A_{14}^{(m_2)} = \frac{1}{2}/D + \frac{1}{2}/A.$$
$$A_5^{(m_2)} = \frac{1}{2}/L + \frac{1}{2}/I, \quad A_{10}^{(m_2)} = \frac{1}{2}/D + \frac{1}{2}/P,$$

3-я группа

$$A_1^{(m_3)} = 1/G, \quad A_6^{(m_3)} = 1/E, \quad A_{11}^{(m_3)} = 1/L,$$
$$A_2^{(m_3)} = 1/G, \quad A_7^{(m_3)} = 1/G, \quad A_{12}^{(m_3)} = 1/D,$$
$$A_3^{(m_3)} = 1/L, \quad A_8^{(m_3)} = 1/D, \quad A_{13}^{(m_3)} = 1/A,$$
$$A_4^{(m_3)} = 1/I, \quad A_9^{(m_3)} = 1/P, \quad A_{14}^{(m_3)} = 1/L.$$
$$A_5^{(m_3)} = 1/R, \quad A_{10}^{(m_3)} = 1/D,$$

Выходные нечеткие множества:

$$B^{(m_1)} = 1/h, \quad B^{(m_2)} = 1/h, \quad B^{(m_3)} = 1/h.$$

Тогда система нечетких правил логического вывода будет иметь следующий вид:

$R^{(1)}$: **if** x_1 есть $A_1^{(m_1)}$ **and** x_2 есть $A_2^{(m_1)}$ **and** … **and** x_{14} есть $A_{14}^{(m_1)}$ **then** y есть B^{m_1},

$R^{(2)}$: **if** x_1 есть $A_1^{(m_2)}$ **and** x_2 есть $A_2^{(m_2)}$ **and** … **and** x_{14} есть $A_{14}^{(m_2)}$ **then** y есть $B^{(m_2)}$,

$R^{(3)}$: **if** x_1 есть $A_1^{(m_3)}$ **and** x_2 есть $A_2^{(m_3)}$ **and** … **and** x_{14} есть $A_{14}^{(m_3)}$ **then** y есть $B^{(m_3)}$.

Существуют два основных способа определения выхода таких систем [4]. В них используется так называемое понятие агрегации правил, т.е. учета суммар-

ного эффекта от использования всех правил. Оператор агрегации **Agg** действует как *s*-норма [3], но допускается использование произвольной *t*-нормы.

Более общей является процедура, которая использует так называемые уровни истинности нечетких правил, т.е. если правило имеет вид

$$\textbf{if } x_1 \text{ есть } A_{i1} \wedge x_2 \text{ есть } A_{i2} \wedge \ldots \wedge x_n \text{ есть } A_{in} \textbf{ then } y \text{ есть } B_i$$

(где x_j, $j = 1, \ldots, n$, — входные лингвистические переменные, y — выходная лингвистическая переменная; A_{ij} и B_i — нечеткие множества; логическая связка $\wedge$ интерпретируется как *t*-норма нечетких множеств), то в случае двух входов x_1 и x_2 процедура выполнения алгоритма будет заключаться в выполнении следующих шагов.

1. Для каждого правила $R^{(i)}$, $i = 1, 2, \ldots, m$, вычисляем его уровень истинности

$$\alpha_i = \min\left[\max_{X_1} (A_1'(x_1) \wedge A_{i1}(x_1)),\ \max_{X_2} (A_2'(x_2) \wedge A_{i2}(x_2))\right].$$

2. Для каждого правила находим индивидуальные выходы

$$B_i'(y) = \min(\alpha_i, B_i(y)).$$

3. Вычисляем агрегатный выход

$$B'(y) = \max(B_1', B_2', \ldots, B_m').$$

Эта процедура называется max-min-процедурой или процедурой логического вывода Мамдани (импликация интерпретируется как операция минимум, агрегация выходов правил — как операция максимум).

Пользуясь последним алгоритмом, определим выход полученной системы нечетких правил. Пусть на вход подается следующая аминокислотная последовательность:

L K V S E G G ***L*** *I R E G Y D P.*

В соответствии с процедурой выполнения алгоритма будем находить уровни истинности правил логического вывода.

1. Уровень истинности первого правила:

$$\alpha_1 = \min[\max(1 \wedge 0,\ 0.5 \wedge 0,\ 0.5 \wedge 0), \max(1 \wedge 0,\ 0.5 \wedge 0.5 \wedge 0),$$
$$\max(1 \wedge 0,\ 0.5 \wedge 0.5 \wedge 0), \max(1 \wedge 0,\ 0.5 \wedge 0,\ 0.5 \wedge 0), \max(1 \wedge 0,\ 0.5 \wedge 0, 0.5 \wedge 0),$$
$$\max(1 \wedge 0,\ 1 \wedge 0), \max(1 \wedge 0.5,\ 0.5 \wedge 0), \max(1 \wedge 0,\ 0.5 \wedge 0,\ 0.5 \wedge 0),$$
$$\max(1 \wedge 0,\ 0.5 \wedge 0,\ 0.5 \wedge 0), \max(1 \wedge 0,\ 0.5 \wedge 0,\ 0.5 \wedge 0), \max(1 \wedge 0,\ 0.5 \wedge 0,\ 0.5 \wedge 0),$$
$$\max(1 \wedge 0,\ 0.5 \wedge 0,\ 0.5 \wedge 0), \max(1 \wedge 0,\ 0.5 \wedge 0,\ 0.5 \wedge 0), \max(1 \wedge 0,\ 0.5 \wedge 0,\ 0.5 \wedge 0),$$
$$\max(1 \wedge 0,\ 0.5 \wedge 0,\ 0.5 \wedge 0), \max(1 \wedge 0.5,\ 0.5 \wedge 0), \max(1 \wedge 0,\ 0.5 \wedge 0,\ 0.5 \wedge 0)] = 0.5.$$

2. Уровень истинности второго правила:

$$\alpha_2 = \min[\max(1 \wedge 0,\ 0.5 \wedge 0,\ 0.5 \wedge 0), \max(1 \wedge 0,\ 0.5 \wedge 0,\ 0.5 \wedge 0),$$
$$\max(1 \wedge 0,\ 0.5 \wedge 0), \max(1 \wedge 0,\ 0.5 \wedge 0,\ 0.5 \wedge 0), \max(1 \wedge 0,\ 0.5 \wedge 0,\ 0.5 \wedge 0),$$
$$\max(1 \wedge 0,\ 0.5 \wedge 0,\ 0.5 \wedge 0), \max(1 \wedge 0,\ 0.5 \wedge 0,\ 0.5 \wedge 0), \max(1 \wedge 0,\ 0.5 \wedge 0,\ 0.5 \wedge 0),$$
$$\max(1 \wedge 0,\ 0.5 \wedge 0,\ 0.5 \wedge 0), \max(1 \wedge 0,\ 0.5 \wedge 0,\ 0.5 \wedge 0), \max(1 \wedge 0,\ 0.5 \wedge 0,\ 0.5 \wedge 0),$$
$$\max(1 \wedge 0,\ 0.5 \wedge 0,\ 0.5 \wedge 0), \max(1 \wedge 0.5,\ 0.5 \wedge 0), \max(1 \wedge 0,\ 0.5 \wedge 0,\ 0.5 \wedge 0)] = 0.5.$$

3. Уровень истинности третьего правила:

$$\alpha_3 = \min[\max(1 \wedge 0,\ 0 \wedge 1), \max(1 \wedge 0,\ 0 \wedge 1),$$
$$\max(1 \wedge 0,\ 0 \wedge 1), \max(1 \wedge 0,\ 0 \wedge 1), \max(1 \wedge 0,\ 0 \wedge 1),$$
$$\max(1 \wedge 0,\ 0 \wedge 1), \max(1 \wedge 0,\ 0 \wedge 1), \max(1 \wedge 1), \max(1 \wedge 0,\ 0 \wedge 1),$$
$$\max(1 \wedge 0,\ 0 \wedge 1), \max(1 \wedge 0,\ 0 \wedge 1), \max(1 \wedge 0,\ 0 \wedge 1),$$
$$\max(1 \wedge 0,\ 0 \wedge 1), \max(1 \wedge 0,\ 0 \wedge 1), \max(1 \wedge 0,\ 0 \wedge 1)] = 1.$$

Вычисляем индивидуальные выходы каждого правила:

$$B'_1(-) = \min(0.5, 0) = 0,\ B'_1(e) = \min(0.5, 0) = 0,\ B'_1(h) = \min(0.5,\ 1) = 0{,}5;$$

$$B'_2(-) = \min(0.5,\ 0) = 0,\ B'_2(e) = \min(0.5,\ 0) = 0,\ B'_2(h) = \min(0.5,\ 1) = 0.5;$$

$$B'_3(-) = \min(1,\ 0) = 0,\ B'_1(e) = \min(1, 0) = 0,\ B'_1(h) = \min(1, 1) = 1.$$

В результате агрегации индивидуальных выходов имеем $B' = 1/h$. Таким образом, вторичная структура остатка $\boldsymbol{L}$ есть h.

ЗАКЛЮЧЕНИЕ

С помощью рассмотренного подхода можно распознавать (предвидеть) вторичную структуру произвольных белков. Чем большее количество обучающих выборок, тем точнее будет распознавание. Учитывая, что наиболее эффективными методами распознавания вторичной структуры белка являются нейросетевые, а также их эквивалентность методам, основанным на системах нечеткого логического вывода, можно утверждать, что с помощью рассматриваемого метода можно достичь достаточно высокой точности распознавания. Кроме того, предложенный метод позволяет учитывать погрешности при определении аминокислотных остатков в обучающих выборках.

СПИСОК ЛИТЕРАТУРЫ

1. Гупал А.М., Сергиенко И.В. Оптимальные процедуры распознавания. — Киев: Наук. думка, 2008. — 382 с.
2. Леск А. Введение в биоинформатику. — М.: Лаборатория знаний, 2009. — 320 с.
3. Рутковская Д., Пилиньский М., Рутковский Л. Нейронные сети, генетические алгоритмы и нечеткие системы. — М.: Телеком, 2006. — 382 с.
4. Провотар А.И., Лапко А.В., Провотар А.А. Нечеткие системы логического вывода и их применение // Кибернетика и системный анализ. — 2013. — № 4. — С. 37–46.

Поступила 17.10.2013