Запросы на сравнение в задаче параметро-эффективной расшифровки булевых функций

Быстрыгова А.В.

В работе рассматривается задача точной параметро-эффективной расшифровки функций из замкнутых классов Поста при помощи запросов на сравнение. Показано, что сложность расшифровки с помощью этих запросов не хуже, а для некоторых классов по порядку лучше, чем таковая для случая запросов на значение.

Ключевые слова: точная расшифровка, параметро-эффективная расшифровка, запросы на значение, запросы на сравнение, замкнутые классы Поста.

1. Введение

Начиная с двадцатого века, одним из популярных направлений исследований в дискретной математике остается изучение задач по расшифровке функций. Эту задачу можно представить как игру между двумя игроками: “учителем“ и “учеником“, в которой учитель выбирает функцию, а ученик должен как можно скорее узнать загаданную учителем функцию, задавая определенные вопросы. При этом ученику лишь известны некоторые параметры, определяющие функцию: число переменных, от которых зависит функция, количество существенных переменных, какое-то свойство, которым она обладает (например, монотонность, количество единиц в векторе значений функции), и так далее.

Многообразие задач в сфере расшифровки функций достигается за счет варьирования модели, в рамках которой проводится расшифровка, классов функций, которые являются целью расшифровки, типов запросов, с помощью которых осуществляется расшифровка. К примеру, в данной работе рассматривается две версии задачи расшифровки, которые отличаются типом запросов к учителю. В первой версии ученик

просит учителя раскрыть значение выбранной функции на каком-то одном наборе значений переменных (запрос на значение), а во второй версии задачи ученик выбирает два набора значений переменных и просит учителя вернуть знак разности значений загаданной функции на этих наборах (запрос на сравнение). Что касается модели расшифровки, то в обеих используется точная модель, то есть, ученик должен полностью восстановить вектор значений загаданной функции.

В работе [1] проведен обзор результатов точной параметро-эффективной расшифровки при помощи запросов на значение для всех замкнутых классов решетки Поста. Насколько целесообразно в этой задаче запросы на значение заменять на запросы на сравнение пока неясно, поскольку в литературе они стали освещаться относительно недавно, когда были впервые введены в [2]. Похожий тип запросов рассматривался в задаче восстановления частичного порядка на множестве [3]. Примеры работ, где изучается сложность расшифровки функций с помощью запросов на сравнение, — [2, 4, 5, 6, 7], но все эти исследования не касались замкнутых классов решетки Поста.

Данная работа посвящена прояснению ситуации, насколько "сильнее" запросы на сравнение относительно запросов на значение в упомянутой выше задаче.

2. Основные понятия и формулировка результатов

Пусть F — некоторый класс функций. Под $F(n,k)$ будем понимать множество функций из F, зависящих от переменных $x_1, x_2, \ldots, x_n$, при этом не более k из которых существенные.

Запросом на сравнение к функции f будем называть пару наборов (a,b), а ответом на него – число $f(a) - f(b)$.

Под *запросом на значение* к функции f будем понимать набор a, а ответом на него будем считать число $f(a)$.

Индекс $_C$ в обозначениях будем приписывать в случае запросов на сравнение и индекс $_V$ — в случае запросов на значение.

Под *алгоритмом расшифровки запросами на сравнение* будем понимать условный эксперимент, который последовательно генерирует запросы на сравнение к функции в зависимости от ответов на предыдущие запросы. Говорим, что *алгоритм расшифровывает функцию* $f \in F(n,k)$,

если ответы на запросы, заданные алгоритмом, однозначно определяют вектор значений функции $f \in F(n,k)$.

Будем говорить, что *класс* $F(n,k)$ *можно расшифровать запросами на сравнение*, если существует алгоритм расшифровки запросами на сравнение, который для любой функции $f \in F(n,k)$ расшифровывает f. Если про класс $F(n,k)$ нельзя сказать, что его можно расшифровать запросами на сравнение, тогда будем, говорить, что *класс* $F(n,k)$ *нельзя расшифровать запросами на сравнение.*

Будем говорить, что *класс* F *можно расшифровать запросами на сравнение*, если для любых целых $n, k (k \leq n)$ можно расшифровать запросами на сравнение класс $F(n,k)$. Если про класс F нельзя сказать, что его можно расшифровать запросами на сравнение, тогда будем, говорить, что *класс* F *нельзя расшифровать запросами на сравнение.*

Под $\varphi_C(A,f)$ будем понимать число запросов на сравнение, требуемое алгоритму (ученику) A, для расшифровки $f \in F(n,k)$.

$T_C(F,n,k)$ — множество всех алгоритмов расшифровки запросами на сравнение функций из $F(n,k)$.

Определим через $\varphi_C(F,n,k) = min_{A \in T_C(F,n,k)} max_{f \in F(n,k)} \varphi_C(A,f)$ минимальное число запросов на сравнение, которое задаст наилучший алгоритм на самой худшей для себя функции из $F(n,k)$.

Аналогично, определяются *алгоритм расшифровки запросами на значение* и $\varphi_V(A,f), T_V(F,n,k), \varphi_V(F,n,k)$.

Далее будем говорить только про *параметро-эффективную расшифровку*, поэтому считаем, что k мало по сравнению с n, то есть $k = o(n)$.

Под $]a[$ будем понимать наименьшее целое, больше или равное a.

Под $|A|$ будем понимать мощность множества A.

Под *набором, сформированным по множеству* A, будем понимать набор, у которого переменные из множества A установлены в 1, остальные – в 0. Под *запросом на сравнение на паре множеств* A, B будем понимать пару наборов, сформированных по множествам A, B.

Будем писать $f \gtrsim g$, если существует такая положительная константа c, что, начиная с некоторого номера n_0, $g(n) \leq c \cdot f(n)$.

Далее напомним обозначения замкнутых классов Поста, при этом классы из "правой" половины решетки Поста заведомо опустим, в силу того, что они являются двойственными к классам из "левой" половины, следовательно задача расшифровки классов из "правой" половины сводится к задаче расшифровки классов из "левой"половины.

C_1 — все функции двузначной логики.

C_2 — все функции, сохраняющие 1.

C_3 — все функции, сохраняющие 0.

C_4 — пересечение C_2, C_3.

A_1 — все монотонные функции.

A_2 — все монотонные функции, сохраняющие 1.

A_3 — все монотонные функции, сохраняющие 0.

A_4 — пересечение A_2, A_3.

F_4^i — класс функций, таких что для любого j $(2 \leq j \leq i)$ верно, что любые j наборов, на которых f обращается в 0, имеют общую нулевую компоненту.

F_3^i — пересечение класса F_4^i и A_2.

F_1^i — пересечение класса F_4^i и C_4.

F_2^i — пересечение класса F_4^i и A_4.

D_3 — все самодвойственные функции.

D_1 — все самодвойственные функции, сохраняющие 0.

D_2 — пересечение классов D_3, F_2^2.

S_1 — все логические суммы вида $x_{i_1} \vee x_{i_2} \vee \ldots \vee x_{i_k}$.

S_3 — S_1 и константа 1.

S_5 — S_1 и константа 0.

S_6 — объединение классов S_3, S_5.

L_1 — все линейные функции.

L_5 — пересечение L_1, D_3.

L_2 — пересечение L_1, C_2.

L_3 — пересечение L_1, C_3.

L_4 — пересечение L_2, L_3, L_5.

$O_9 = \{1, 0, x, \bar{x}\}$.

$O_6 = \{0, 1, x\}$.

$O_4 = \{x, \bar{x}\}$.

$O_5 = \{1, x\}$.

$O_8 = \{0, x\}$.

$O_1 = \{x\}$.

Теорема 1. *Класс $F(n, k)$ расшифровать запросами на сравнение нельзя тогда и только тогда, когда обе константные функции $0, 1 \in F(n, k)$.*

Непосредственно из этой теоремы получаем следующее следствие.

Следствие 1. *Если $F \in \{C_1, A_1, L_1, S_6, O_9, O_6\}$, то расшифровать запросами на сравнение класс F невозможно. Если $F \in \{C_2, C_3, C_4, A_2, A_3, A_4, F_4^i, F_3^i, F_1^i, F_2^i, D_1, D_2, S_1, S_3, S_5, L_2, L_3, L_4\}$ или $F \in \{O_8, O_5, O_1, O_4, D_3, L_5\}$, то класс F можно расшифровать.*

Теорема 2. *Сложность расшифровки запросами на сравнение функций из классов* $C_2, C_3, C_4, A_2, A_3, A_4, F_4^i, F_3^i, F_1^i, F_2^i, D_1, D_2$ *и классов* $S_1, S_3, S_5, L_2, L_3, L_4, O_8, O_5, D_3, L_5$ *не хуже, чем сложность расшифровки запросами на значение.*

Теорема 3. *Справедливы следующие соотношения:*

1) $\varphi_C(O_1, n, 1) =]\log_3 n[\leq]\log_2 n[= \varphi_V(O_1, n, 1)$,

2) $\varphi_C(O_4, n, 1) \leq]\log_3 n[+ 1 \leq]\log_2 n[+ 1 = \varphi_V(O_4, n, 1)$.

Теорема 4. *Если при расшифровке запросами на сравнение разрешить ровно один запрос на значение, тогда классы* $C_1, A_1, L_1, S_6, O_9, O_6$ *можно расшифровать со сложностью не более, чем на один запрос больше, чем при расшифровке, в которой разрешены только запросы на значение.*

3. Доказательство теорем

Далее приведем доказательство теоремы 1.

Доказательство. Докажем достаточность. Обе константные функции на любом запросе на сравнение возвращают значение 0, поэтому распознать, какая из них загадана невозможно.

Докажем необходимость. От противного. Пусть $\{0, 1\} \not\subseteq F(n, k)$.

Поскольку класс $F(n, k)$ расшифровать нельзя, то какой бы алгоритм расшифровки $A \in T_C(F, n, k)$ не взять, то найдется функция $f \in F(n, k)$, которую этот алгоритм не сможет расшифровать, то есть нельзя отличить ее от какой-то другой функции из класса $F(n, k)$. Рассмотрим следующий алгоритм Q. В качестве запросов на сравнение возьмем множество всех пар $(a, b), a, b \in \{0, 1\}^n$. Зафиксируем любую функцию $f \in F(n, k)$.

1) Если $f \in \{0, 1\}$, то ответы на все запросы будут равны 0, что отличает ее от остальных неконстантных функций, для которых среди ответов на запросы обязательно встретится хотя бы одна 1.

2) Если $f \notin \{0, 1\}$, то среди ответов на запросы встречается хотя бы одна 1, что отличает эту функцию от константной. Рассмотрим любую отличную от f функцию $g \in F(n, k), g \notin \{0, 1\}$. Функции f, g отличаются, значит, существует набор a такой, что $f(a) \neq g(a)$. Возможны два случая:

- существует набор b такой, что $f(b) = g(b)$, тогда на запросе (a, b) функции f, g вернут разные ответы;
- для любого набора b верно неравенство $f(b) \neq g(b)$, тогда на запросе (c, d), где $f(c) = 0, f(d) = 1$, функция f вернет ответ -1, а функция g вернет ответ 1.

Следовательно, алгоритм Q расшифровывает класс $F(n, k)$, получили противоречие.

□

Утверждение 1. *Если для некоторого набора a, некоторого числа $b \in \{0, 1\}$ для любой $f \in F(n, k)$ известно, что $f(a) = b$, и $\{0, 1\} \not\subseteq F(n, k)$, тогда $\varphi_C(F, n, k) \leq \varphi_V(F, n, k)$.*

Доказательство. В силу теоремы 1, класс $F(n, k)$ можно расшифровать запросами на сравнение, поэтому дальнейшие рассуждения имеют смысл.

Рассмотрим алгоритм расшифровки A запросами на значение, на котором достигается минимум выражения $max_{f \in F(n,k)} \varphi_V(A, f)$. По алгоритму A построим алгоритм расшифровки B запросами на сравнение следующим образом: первая компонента каждого запроса на сравнение есть набор из алгоритма A, а вторая компонента запроса на сравнение – это набор a. Тогда для любой $f \in F(n, k)$ верно соотношение $\varphi_C(B, f) = \varphi_V(A, f)$. Отсюда следует равенство $max_{f \in F(n,k)} \varphi_C(A, f) = max_{f \in F(n,k)} \varphi_V(A, f)$. Учитывая, что в $T_C(F, n, k)$ помимо алгоритмов расшифровки, предложенных выше, возможно имеются еще какие-то, получаем неравенство из утверждения. □

Утверждение 2. *Если $F(n, k) \subseteq D_3(n, k)$, тогда $\varphi_C(F, n, k) \leq \varphi_V(F, n, k)$.*

Доказательство. Рассмотрим алгоритм расшифровки A запросами на значение, на котором достигается минимум выражения $max_{f \in F(n,k)} \varphi_V(A, f)$. По алгоритму A построим алгоритм расшифровки B запросами на сравнение следующим образом: первая компонента каждого запроса на сравнение есть набор из алгоритма A, а вторая компонента запроса на сравнение – это противоположный ему набор. Ответ на любой такой запрос на сравнение однозначно восстановит значение функции на обоих наборах в силу того, что наборы противоположные, а функция самодвойственная. Поэтому для любой

$f \in F(n,k)$ верно соотношение $\varphi_C(B,f) = \varphi_V(A,f)$. Отсюда следует $max_{f\in F(n,k)}\varphi_C(A,f) = max_{f\in F(n,k)}\varphi_V(A,f)$. Учитывая, что в $T_C(F,n,k)$ помимо алгоритмов расшифровки, предложенных выше, возможно имеются еще какие-то, получаем неравенство из утверждения. □

Утверждение 3. *(Нижняя мощностная оценка) Если* $\{0,1\} \not\subseteq F(n,k)$, *то* $\varphi_C(F,n,k) \geq]\log_3 |F(n,k)|[$.

Доказательство. В силу теоремы 1, класс $F(n,k)$ можно расшифровать запросами на сравнение, поэтому дальнейшие рассуждения имеют смысл.

Доказательство аналогично случаю расшифровки запросами на значение.

Пусть задано x запросов на сравнение, на каждый запрос возможны 3 варианта ответа, следовательно, всего возможных векторов ответов не более 3^x ("не более" в силу того, что какие-то комбинации ответов на запросы могут быть между собой не согласованы, то есть не определяют функцию из заданного класса).

Необходимо задать такое количество запросов, чтобы однозначно восстановить функцию, поэтому должно выполняться следующее неравенство $3^x \geq |F(n,k)|$. Отсюда следует неравенство утверждения. □

Утверждение 4. *Если известно, что среди множества переменных* $A = \{x_{i_1}, x_{i_2}, \ldots, x_{i_q}\}, 1 \leq i_1 < i_2 < \ldots < i_q \leq n$, *нечетное число существенных переменных функции* $f(x_1, x_2, \ldots, x_n) \in L_3$. *Тогда одну существенную переменную можно найти, задав не более* $]\log_3 q[$ *запросов на сравнение.*

Доказательство. Шаг 1. Если $|A| = 1$, тогда СТОП, переменная из множества и есть существенная. Иначе, разделить множество A на непересекающиеся множества A_1, A_2, A_3, отличающиеся по мощности не более, чем на один элемент. Перейти к шагу 2.

Шаг 2. Запросить значение на сравнение на множествах A_1, A_2. Если значение равно 1, тогда положить $A = A_1$, перейти к шагу 1. Если значение равно -1, тогда положить $A = A_2$, перейти к шагу 1. Иначе, положить $A = A_3$, перейти к шагу 1.

Если на шаге 2 ответ на запрос равен 0, то значит, либо в обоих A_1, A_2 нечетное число существенных из множества A и значение функции на наборах, сформированных по каждому из множеств A_1, A_2 равно 1, либо в обоих A_1, A_2 четное число существенных из множества A и значение

функции на наборах, сформированных по каждому из множеств A_1, A_2 равно 0. В обоих случаях, суммарно в A_1, A_2 содержится четное число существенных переменных, значит нечетное число существенных содержится в A_3.

После каждого шага 2 мощность множества A уменьшается как минимум в 3 раза.

□

Далее приведем доказательство теоремы 2.

Доказательство. Если $F \in \{C_2, C_3, C_4, A_2, A_3, A_4, F_4^i, F_3^i, F_1^i, F_2^i\}$ или $F \in \{D_1, D_2, S_1, S_3, S_5, L_2, L_3, L_4, O_8, O_5\}$, то согласно утверждению 1 получаем неравенство $\varphi_C(F, n, k) \leq \varphi_V(F, n, k)$.

Если $F \in \{D_3, L_5\}$, то в силу утверждения 2 получаем неравенство $\varphi_C(F, n, k) \leq \varphi_V(F, n, k)$.

□

Приведем доказательство теоремы 3.

Доказательство. Соотношение $\varphi_C(O_1, n, 1) =]\log_3 n[$ следует из утверждений 3 и 4. Учитывая соотношение $\varphi_V(O_1, n, 1) =]\log_2 n[$ из [8], получаем первый пункт теоремы.

Для получения верхней оценки для $\varphi_C(O_4, n, 1)$ предлагается следующий алгоритм расшифровки: узнаем значение на запросе $(0 \ldots 0, 1 \ldots 1)$, если ответ равен -1, то загадана функция $x_i, 1 \leq i \leq n$, иначе загадана функция $\bar{x_i}, 1 \leq i \leq n$. В случае функции x_i применяем алгоритм расшифровки, описанный в доказательстве утверждения 4, в случае функции $\bar{x_i}$ применяем похожий алгоритм. В результате, получаем неравенство $\varphi_C(O_4, n, 1) \leq 1 +]log_3 n[$. Учитывая соотношение $\varphi_V(O_4, n, 1) =]\log_2 n[+ 1$ с [1], получаем второй пункт теоремы.

□

Наконец, приведем доказательство теоремы 4.

Доказательство. В самом начале расшифровки сделаем один запрос на значение — запросим значение функции на любом наборе q. Далее в силу того, что известно значение загаданной функции на одном наборе, воспользуемся аналогом доказательства утверждения 1: зафиксируем “наилучший алгоритм расшифровки“ A соответствующего класса запросами на значение, и вместо каждого его запроса t будем подавать запрос на

сравнение (q, t), тем самым мы узнаем значение на всех наборах алгоритма A, ведь точно известно значение функции на наборе q. Поскольку алгоритм A расшифровывал функцию, то и полученный алгоритм расшифровки запросами на сравнение и одним запросом на значение восстанавливает ее. □

4. Благодарность

Автор выражает благодарность научному руководителю — д.ф-м.н., профессору Э. Э. Гасанову за постановку задачи и помощь в работе.

Список литературы

[1] А. В. Быстрыгова, "Параметро-эффективная расшифровка булевых функций из замкнутых классов Поста", *Дискрет. матем.*, **31**:2 (2019), 34–58.

[2] Гасанов Э.Э., "Расшифровка линейных функций ранжирования", *Материалы XI Международного семинара "Дискретная математика и ее приложения посвященного 80-летию со дня рождения академика О.Б.Лупанова (Москва, 18-23 июня 2012 г.)*, Изд-во мех-мат фак-та МГУ, 2012, 332–334.

[3] А. А. Абдель Маджид, "О сложности восстановления частичного порядка", *Интеллектуальные системы*, **20**:4 (2016), 5–10.

[4] Гасанов Э. Э., Ниязова З. А., "Расшифровка арифметических сумм малого числа монотонных конъюнкций", *Материалы XI Международного семинара Дискретная математика и ее приложения (Москва, 18-23 июня 2012 г.)*, Изд-во механико-математического ф-та МГУ, Москва, 2012, 335–337.

[5] Ниязова З. А., "Расшифровка арифметических сумм монотонных конъюнкций", *Интеллектуальные системы. Теория и приложения*, **19**:4 (2015), 169–195.

[6] Хегай С.И., "Расшифровка полиномиальных функций ранжирования", *Интеллектуальные системы*, **19**:1 (2015), 213–230.

[7] Быстрыгова А.В., "Письмо в редакцию по поводу статьи З.А.Ниязовой "Расшифровка арифметических сумм монотонных конъюнкций"", *Интеллектуальные системы*, **22**:4 (2018), 107–109.

[8] R. Uehara, K. Tsuchida, I. Wegener, "Optimal Attribute-Efficient Learning Of Disjunction, Parity, And Threshold Functions", *EuroCOLT '97 Proceedings of the Third European Conference on Computational Learning Theory*, 1997.

Using comparation queries in attribute-efficient learning of Boolean functions
Bistrigova A.V.

We consider the problem of exact attribute-efficient learning functions of Post's closed classes with the help of comparation queries. Here, we show that the complexity of learning by comparation queries is not worse than by membership queries. Particularly, for some classes, if we use comparation queries, we get better value of complexity function.

Keywords: exact learning, attribute-efficient learning, membership queries, comparation queries, Post's closed classes.