Код Рида-Маллера

Илья Коннов

Факультет компьютерных наук Высшая Школа Экономики

12 марта 2022 г.

Код Рида-Маллера

2022-03-

- 1. Существует три различных варианта этого доклада:

 - Краткая презентация, которую несложно рассказать, но может быть сложно понять (ReedMuller-trans.pdf).
 Более длинная презентация с ценными комментариями, дополнительными доказательствами и интересными фактами (ReedMuller-slides.pdf).
 Текстовая статья со всем содержимым длинной презентации, комментариями на своих местах, а также бонусным приложением с более подробным описанием алгоритма (ReedMuller-article.pdf).

Их все можно посмотреть здесь: https://sldr.xyz/ReedMuller/

По любым вопросам: r-m@sldr.xyz или t.me/iliago или vk.com/iliago

Введение

Код описан Дэвидом Маллером (автор идеи) и Ирвингом Ридом (автор метода декодирования) в сентябре 1954 года.

Обозначается как $\mathrm{RM}(r,m)$, где r — ранг, а 2^m — длина кода. Кодирует сообщения длиной $k=\sum_{i=0}^r C_m^i$ при помощи 2^m бит. Традиционно, считается что коды бинарные и работают над битами, т.е. \mathbb{F}_2 .

Соглашение: сложение векторов $u,v\in\mathbb{F}_2^n$ будем обозначать как

 $u\oplus v=(u_{1}+v_{1},u_{2}+v_{2},...,u_{n}+v_{n}).$



Булевы функции и многочлен Жегалкина

Всякую булеву функцию можно записать при помощи таблицы истинности:

Или при помощи многочлена Жегалкина:

$$f(x,y) = xy + x + y + 1$$



Многочлены Жегалкина

В общем случае, многочлены будут иметь следующий вид:

$$f(x_1,x_2,...,x_m) = \sum_{S \subseteq \{1,...,m\}} c_S \prod_{i \in S} x_i$$

Например, для m=2: $f(x_1,x_2)=c_{12}\cdot x_{\{1\}}x_2+c_{\{2\}}\cdot x_2+c_{\{1\}}\cdot x_1+c_{\varnothing}\cdot 1$ Всего $n=2^m$ коэффициентов для описания каждой функции.



Функции небольшой степени

Рассмотрим функции, степень многочленов которых не больше r:

$$\{f(x_1,x_2,...,x_m)\mid \deg f\leq r\}$$

Каждую можно записать следующим образом:

$$f(x_1,x_2,...,x_m) = \sum_{\substack{S \subseteq \{1,...,m\}\\|S| \le r}} c_S \prod_{i \in S} x_i$$

В каждом произведении используется не больше r переменных. Сколько тогда всего коэффициентов используется?

$$k = C_m^0 + C_m^1 + C_m^2 + \ldots + C_m^r = \sum_{i=0}^r C_m^i$$

Код Рида-Маллера —Введение 2022-03 ∟Функции небольшой степени



- 1. Замечу, что при $S=\varnothing$, мы считаем, что $\prod_{i\in S} x_i=1$, таким образом всегда появляется
- 2. Если говорить несколько проще, то для составления многочленов мы сложим сначала одночлены $(x+y+z+\ldots)$, затем произведения одночленов $(xy+yz+xz+\ldots)$ и т.д. вплоть до r множителей (поскольку мы работаем в поле \mathbb{F}_2 , здесь нету x^2,y^2,z^2 , т.к. $a^2=a$). Тогда легко видеть, почему k именно такое: мы складываем все возможные перестановки сначала для 0 переменных, потом для одной, двух, и так вплоть до r (не не больше, ведь $\deg f \leq r$).

Код Рида-Маллера Кодирование

∟Идея кодирования



- 1. Их 2^m , поскольку рассматриваем многочлены только над \mathbb{F}_2 от m переменных. 2. Вектор значений обозначается $\mathrm{Eval}(f)$ столбец таблицы истинности, содержащий значения функции. Имеет смысл только при зафиксированном порядке строк в таблице. У меня он везде самый обычный, как в примере выше.

Код Рида-Маллера —Кодирование **∟**Пример



- 1. Здесь и далее я для краткости и удобства записываю битовые векторы не как $\begin{pmatrix} 1 & 0 & 0 & 1 \end{pmatrix}$, а
- как 1001 при помощи нескучного шрифта.
 2. Для кодирования очень важно понимать, как именно биты сообщения ставятся в соответствие коэффициентам многочлена. Поэтому давайте введём соглашение: если упорядочить элементы множества у каждого коэффициента по возрастанию, то коэффициенты сортируются в лексиографическом порядке: $c_{1,2}$ раньше $c_{1,3}$, поскольку 2<3 и $c_{2,3}$ раньше $c_{3,4}$, поскольку 2<3. Пример для m=4:

$$\begin{split} f(x_1,x_2,x_3,x_4) &= c_{\{1,2,3,4\}} x_1 x_2 x_3 x_4 \\ &+ c_{\{1,2,3\}} x_1 x_2 x_3 + c_{\{1,2,4\}} x_1 x_2 x_4 + c_{\{1,3,4\}} x_1 x_3 x_4 + \\ &+ c_{\{2,3,4\}} x_2 x_3 x_4 \\ &+ c_{\{1,2\}} x_1 x_2 + c_{\{1,3\}} x_1 x_3 + c_{\{1,4\}} x_1 x_4 + c_{\{2,3\}} x_2 x_3 + \\ &+ c_{\{1,2\}} x_1 x_2 + c_{\{2,4\}} x_2 x_4 + c_{\{3,4\}} x_3 x_4 \\ &+ c_{\{1\}} x_1 + c_{\{2\}} x_2 + c_{\{3\}} x_3 x_4 + c_{\{4\}} x_4 + c_6 \end{split}$$

Также можно кодировать множества при помощи битов, используя отношение $x\in A\Leftrightarrow v_x=1$ (нумерация битов слева направо, начиная с единицы), где свойство остортированности сохраняется и хорошо видно (но только в пределах группы мнономов одной степени):

$$\begin{split} f(x_1, x_2, x_3, x_4) &= c_{1111} x_1 x_2 x_3 x_4 \\ &+ c_{1116} x_1 x_2 x_3 + c_{1161} x_1 x_2 x_4 + c_{1011} x_1 x_3 x_4 + c_{0111} x_2 x_3 x_4 \\ &+ c_{1166} x_1 x_2 + c_{1016} x_1 x_3 + c_{1061} x_1 x_4 + c_{0116} x_2 x_3 + \\ &+ c_{1061} x_2 x_4 + c_{0011} x_3 x_4 \end{split}$$

Идея кодирования

Пусть каждое сообщение (длины k) — коэффициенты многочлена от m

переменных степени не больше r. . Тогда мы можем его представить при помощи 2^m бит, подставив все

возможные комбинации значений переменных. Таким образом получим таблицу истинности, из которой позднее сможем восстановить исходный многочлен, а вместе с ним и сообщение.

Зафиксировав в таблице порядок строк, можно выделить вектор значений, который и будет кодом.

Пример

r = 1 (степень многочлена), m = 2 (переменных). Это RM(1,2).

- \blacksquare Тогда наш многочлен: $f(x_1,x_2)=c_{\{2\}}x_2+c_{\{1\}}x_1+c_{\varnothing}.$
- lacksquare Сообщение: 011, тогда $f(x_1,x_2)=0+x_1+1.$
- Подставим всевозможные комбинации:

x_1	x_2	$f(x_1,x_2)$
0	0	1
0	1	1
1	0	0
1	1	0

■ Получили код: $\mathrm{Eval}(f) = 1100$.



Декодирование когда потерь нет

■ Мы получили код: 1100 ■ Представим таблицу истинности.

■ Подстановками в $f(x_1,x_2) = c_2x_2 + c_1x_1 + c_0$

получим СЛАУ.

0 1 1 0 $c_0 = 1$ c_2 $+ c_0 = 1$ $c_0^{"} = 0$ c_1 +

 $x_2 \, \Big\rfloor \, f(x_1,x_2)$

 $\blacksquare \ c_{\{1\}} = 1, c_{\{2\}} = 0, c_{\varnothing} = 1$, исходное сообщение: 011.

Код Рида-Маллера —Кодирование

—Декодирование когда потерь нет



1. Теперь покажем, как можно декодировать когда потерь нет. Этот пример — продолжение предыдущего.

Коды 0-го порядка

Для случая $\mathrm{RM}(0,m)$ нужна функция от m аргументов, степени не выше 0. $\quad \blacksquare \ f(x_1,x_2,...,x_m)=0$

 $\quad \blacksquare \ g(x_1,x_2,...,x_m)=1$

Таблица истинности:

	x_1	x_2		x_m	$f(x_1,,x_m)$	$g(x_1,,x_m)$
	(0	0		0	0	1
\mathfrak{I}^m	$\begin{cases} 0 \\ 0 \end{cases}$	0		1	0	1
2	ì		٠.		:	:
	1	1		1	0	1

Вывод: это 2^m -кратное повторение символа

- Сообщение 0 даст код 00...0
- Сообщение 1 даст код 11...1

Код Рида-Маллера —Кодирование

∟Коды 0-го порядка



- 1. Отдельно стоит рассмотреть вариант кода при r=0, он нам в будущем пригодится для
- доказательств.
 2. Таких функций существует всего лишь две, поскольку мы можем влиять лишь на свободный
- член. Все остальные коэффициенты обнуляются из-за требования $\deg f \leq 0$. 3. Здесь число строк, как и в любой другой таблице истинности, равно 2^m , а колонки со значениями никак не зависят от аргументов функций. Получается две колонки одна с нулями, другая с единицами.

Коды m-го порядка

Есть m переменных, и мы рассматриваем многочлены $f \in \mathbb{F}_2[x_1,...,x_m] : \deg f \leq m$, т.е. все возможные.

Для $\widetilde{\mathrm{RM}}(m,m)$ мы используем все доступные коэффициенты многочлена для кодирования сообщения.

Тогда нет избыточности: $k = \sum_{i=0}^m C_m^i = 2^m = n$ – длина сообщения равна длине кода.

Чем меньше порядок кода r, тем больше избыточность.

Код Рида-Маллера —Кодирование \sqsubseteq Коды m-го порядка

1. Есть ещё один тривиальный случай, когда m=r.

Доказательство линейности

Пусть C(x) кодирует сообщение $x \in \mathbb{F}_2^k$ в код $C(x) \in \mathbb{F}_2^m$.

 $C(x) = (p_x(a_i) \mid a_i \in \mathbb{F}_2^m)$

где $p_x(a_i)$ — соответствующий сообщению x многочлен. Причём p_x берёт в качестве своих коэффициентов биты из x. Поскольку многочлены степени не выше r образуют линейное пространство, то

 $p_{(x \oplus y)} = p_x + p_y.$ Тогда:

 $C(x\oplus y)_i=p_{(x\oplus y)}(a_i)=p_x(a_i)+p_y(a_i)=C(x)_i+C(y)_i$

т.е. $\forall x,y \quad C(x\oplus y) = C(x) + C(y)$, ч.т.д.

Код Рида-Маллера └-Свойства кода 2022-03-

—Доказательство линейности

еде $y_c(a_s)$ — экспектицииций кообщинии и минисими. Причём y_s бирёт и минисим моне моне монероциямия биль им и. По

- 1. Хотим показать, что этот код является линейным, т.е. что его кодовые слова образуют линейное пространство, и у нас есть изоморфизм из пространства сообщений (\mathbb{F}_2^k) в пространство слов $(\mathbb{F}_2^{n_1})$. Для этого необходимо немного формализовать всё описанное раньше.
- 2. Пояснение: перебираем все векторы a_1 (2^m штук), подставляем каждый в p_x в качестве переменных и таким образом получаем вектор значений (длины 2^m). Именно он и называется
- 3. Напомню, что базис пространства многочленов выглядит примерно так: 1, x, y, z, xy, yz, xz(для трёх переменных, степени не выше 2). Чтобы преобразовать сообщение в многочлен, мы берём каждый бит сообщения и умножаем
- его на соответствующий базисный вектор. Очевидно, такое преобразование будет изоморфизмом. Именно поэтому $p_{(x\oplus y)}=p_x+p_y$. Обратите внимание, что сообщение x это не просто число (\mathbb{Z}_{2^k}) и мы рассматриваем его биты, а реально вектор битов (\mathbb{Z}_k^k) . У него операция сложения побитовая
- операция Сложеняя поотговая. 4. Здесь я использую запись $C(x)_i$ для i-го элемента вектора C(x). Поскольку i произвольное, то и весь вектор получился равен. Таким образом, этот код действительно линейный и к нему применимы уже известные теоремы!

Faculty o	
Computer	
HSt University	

Последствия линейности

 \blacksquare Существует порождающая матрица G.

$$C(x) = x_{1 \times k} G_{k \times n} = c_{1 \times n}$$

2 Минимальное расстояние будет равно минимальному весу Хемминга среди всех кодов.

$$d = \min_{c \in C} w(c)$$

Корректирующая способность:

$$t = \left| \frac{d-1}{2} \right|$$



└─Последствия линейности

- 1. Так можно кодировать сообщения x в коды c. Но искать её мы не будем, обойдёмся одними
- многочленами, это интереснее. Вес Хэмминга вектора количество в нём ненулевых элементов.
- Доказательство очень просто: минимальное расстояние вес разности каких-то двух
 различных кодов, но разность двух кодов тоже будет кодом, т.к. мы в линейном пространстве.
 Значит достаточно найти минимальный вес, но не учитывая нулевой вектор, т.к. разность
- равна нулю тогда и только тогда, когда коды равны. 4. Однако мы ещё не знаем как выглядят наши коды (как выглядят таблицы истинности функций степени не больше r?). А значит не можем ничего сказать про минимальное расстояние.



Конструкция Плоткина: многочлены

Хотим понять как выглядят кодовые слова.

- lacktriangle Код вектор значений функции $f(x_1,...,x_m)\in \mathrm{RM}(r,m)$, причём
- \blacksquare Разделим функцию по $x_1{:}\ f(x_1,...,x_m) = g(x_2,...,x_m) + x_1h(x_2,...,x_m).$
- $lacksymbol{\blacksquare}$ Заметим, что $\deg f \leq r$, а значит $\deg g \leq r$ и $\deg h \leq r-1$.

Код Рида-Маллера Свойства кода

—Конструкция Плоткина

Конструкция Плоткина: многочлены

- 1. Порядок очевидно не больше r, потому что это условие для включения в пространство кодов
- 2. Теперь у нас есть две функции от меньшего числа аргументов. Очевидно, так можно сделать всегда, когда т > 1.

Конструкция Плоткина: таблица истинности

 $\mbox{ Ранее: } f(x_1,...,x_m) = g(x_2,...,x_m) + x_1 h(x_2,...,x_m).$

 \blacksquare Заметим, что таблица истинности f состоит из двух частей: при $x_1=0$ и при $x_1=1$.

 $\mathrm{Eval}(f) = \left(\frac{\mathrm{Eval}^{[x_1=0]}(f)}{\mathrm{Eval}^{[x_1=1]}(f)}\right)$

- lacktriangle Причём $\operatorname{Eval}^{[x_1=0]}(f) = \operatorname{Eval}(g)$, а $\operatorname{Eval}^{[x_1=0]}(f) \oplus \operatorname{Eval}^{[x_1=1]}(f) = \operatorname{Eval}(h)$.
- \blacksquare Таким образом, $\mathrm{Eval}(f) = (\mathrm{Eval}(g) \mid \mathrm{Eval}(g) \oplus \mathrm{Eval}(h)).$

Код Рида-Маллера —Свойства кода —Конструкция Плоткина

Конструкция Плоткина: таблица истинности

npa $x_i = 1$. $\operatorname{End}(f) = \frac{\left(\operatorname{End}^{(r)-10}(f)\right)}{\left(\operatorname{End}^{(r)-10}(f)\right)}$ • Equation $\operatorname{End}^{(r)-10}(f) = \operatorname{End}(g)$, • Equation $\operatorname{End}^{(r)-10}(f) \oplus \operatorname{End}(g)$

- 1. Теперь рассмотрим те же функции, но со стороны их таблиц истинности. Нам же интересны именно коды, а они как раз очень тесно связаны с этими таблицами. 2. Про обозначения: $\mathrm{Eval}(f) = \mathrm{Taблицa}$ для всей функции (вектор значений, если точнее), $\mathrm{Eval}^{|x_1=0|}(f) = \mathrm{Kycok}$ таблица для всей функции (вектор значений, если точнее), $\mathrm{Eval}^{|x_1=0|}(f) = \mathrm{Kycok}$ таблицы при $x_1=0$. Они нам после этого доказательства больше не понадобятся. 3. Это всё следует из ранее полученного утверждения. Если мы подставим $x_1=0$, то останется только $g = \mathrm{nepsee}$ развенство очещдно. Если же мы рассмотрим $\mathrm{Eval}^{|x_1=1|}(f)$, то получим $\mathrm{Eval}(g+h)$, но если туда прибавить ещё раз $\mathrm{Eval}(g)$, то останется только $\mathrm{Eval}(h)$ (поскольку 1+1=0 в \mathbb{F}_2) получили второе разенство. 4. Палочка по центоту конкатенация векторов.
- 4. Палочка по центру конкатенация векторов.

Конструкция Плоткина: вывод

Если дана $f(x_1,...,x_m)$, причём $\deg f \leq r$, то можно её разделить:

$$f(x_1,...,x_m) = g(x_2,...,x_m) + x_1 h(x_2,...,x_m)$$

Также известно, что $\operatorname{Eval}(f) = (\operatorname{Eval}(g) \mid \operatorname{Eval}(g) \oplus \operatorname{Eval}(h)).$

Заметим, что $\operatorname{Eval}(f)$ – кодовое слово (как и для g и h). Тогда: $c = \operatorname{Eval}(f) \in \operatorname{RM}(r, m)$

огда:
$$c=\mathrm{Eval}(f)\in\mathrm{RM}(r,m)$$
 (т.к. $\deg f\leq r$)
$$u=\mathrm{Eval}(g)\in\mathrm{RM}(r,m-1)$$
 (т.к. $\deg g\leq r$)

$$u = \operatorname{Eval}(g) \in \operatorname{RM}(r, m-1)$$
 (т.к. $\deg g \le r$) $v = \operatorname{Eval}(h) \in \operatorname{RM}(r-1, m-1)$ (т.к. $\deg h \le r-1$)

Код Рида-Маллера — Свойства кода Конструкция Плоткина Конструкция Плоткина: вывод

- 1. Теперь собираем всё это в одно важное утверждение. 2. Причём мы уже знаем, что $\deg g \le r$ и $\deg h \le r-1$, если $\deg f \le r$ 3. Напомню, что $\mathrm{RM}(r,m)$ включает в себя **все** функции (их таблицы истинности, если точнее) от m аргументов и степени не выше r. Очевидно, наши годятся.



Конструкция Плоткина

Теорема

Для всякого кодового слова $c\in\mathrm{RM}(r,m)$ можно найти $u\in\mathrm{RM}(r,m-1)$ и $v \in \mathrm{RM}(r-1,m-1)$, такие что $c = (u \mid u+v).$

Код Рида-Маллера — Свойства кода Конструкция ПлоткинаКонструкция Плоткина

1. Что здесь важно отметить — оба наших новых кодовых слова u,v получились «меньше», чем

Минимальное расстояние

 $d = \min_{c \in C, c \neq 0} w(c)$

Предположим, что $d=2^{m-r}$ и докажем по индукции.

Хотим найти минимальное расстояние для кода $\mathrm{RM}(r,m)$

База: $\mathrm{RM}(0,m)$ — единственный бит повторён 2^m раз. Очевидно, $w(\underbrace{11...1}) = 2^m = 2^{m-0} \geq 2^{m-r}.$

Гипотеза: Если $v \in \text{RM}(r-1,m-1)$, то $w(v) \ge 2^{m-r}$.

f f Uаг: Хотим доказать для $c\in {
m RM}(r,m).$

$$\begin{split} w(c) &\stackrel{(1)}{=} w\big((u \mid u \oplus v)\big) \stackrel{(2)}{=} w(u) + w(u \oplus v) \geq \\ &\stackrel{(3)}{\geq} w(u) + (w(v) - w(u)) = w(v) \stackrel{IH}{\geq} 2^{m-r} \blacksquare \end{split}$$

Код Рида-Маллера —Свойства кода —Минимальное расстояние -Минимальное расстояние

1. Случай $\mathrm{RM}(0,m)$ мы разбирали раньше, но я напомню. Здесь длина сообщения равна $k=\sum_{i=0}^r C_m^i=C_m^0=1$, а длина кода $n=2^m$. Причём мы просто берём один бит и повторяем его 2^m раз (в таблице истинности). Замечу, что не рассматриваю второй случай $w(\theta0...\theta)$, поскольку он нам не нужен для расчёта минимального расстояния. Вариант с нулевым вектором явно выкидывается, см. определение d

2. Теперь немного объяснений. Переход (1): используем конструкцию Плоткина, чтобы разбить c на конкатенацию двух одовых слов поменьше.

кодовых слов поменьше. Переход (2): $w((x\mid u))=w(x)+w(y)$. Вес это всего лишь число ненулевых элементов, поэтому нет разницы как мы будем группировать части вектора. Переход (3): $w(u\not v)\geq w(v)-w(u)$. Если у нас в v стоит w(v) бит, то прибавив к нему u, мы сможем изменить (обнулить) не больше w(u) бит. Возможно появится больше единиц, но нас интересует нижняя граница. Переход (IH): предположение индукции в чистом виде.

Свойства и параметры

Для бинарного кода $\mathrm{RM}(r,m)$:

- - $r \leq m$ \blacksquare Длина кода: 2^m
 - Длина сообщения: $k = \sum_{i=0}^{r} C_m^i$ Минимальное расстояние: $d = 2^{m-r}$

 - lacktriangle Корректирующая способность: $t=2^{m-r-1}-1$
 - lacktriangle Существует порождающая матрица G для кодирования
 - \blacksquare Проверочная матрица H совпадает с порождающей для $\mathrm{RM}(m-r-1,m)$

Код Рида-Маллера -Свойства кода □Параметры

Свойства и параметры

- 1. Теперь можно подвести итоги исследования свойств.
 2. , поскольку $t = \lfloor \frac{d-1}{2} \rfloor = \lfloor \frac{2^{m-r}}{2} \frac{1}{2} \rfloor = \lfloor 2^{m-r-1} 0.5 \rfloor = 2^{m-r-1} 1$
 3. , она позволяет делать так: C(x) = xG. Но я, как обычно, её избегаю. Рекомендую почитать «Коды Рида-Маллера: Примеры исправления ошибок», если интересно.
 4. , но это я это доказывать не собираюсь. Однако доказательство можно найти в «Reed-Muller Codes: Theory and Algorithms», раздел Duality.

Возможные варианты

m	0	1	2	3	4
1	k = 1 $n = 2$ $t = 0$	k = 2 $n = 2$ $t = 0$	_	_	_
2	k = 1 $n = 4$ $t = 1$	k = 3 $n = 4$ $t = 0$	k = 4 $n = 4$ $t = 0$	_	_
3	k = 1 $n = 8$ $t = 3$	k = 4 $n = 8$ $t = 1$	k = 7 $n = 8$ $t = 0$	k = 8 $n = 8$ $t = 0$	_
4	k = 1 $n = 16$ $t = 7$	k = 5 $n = 16$ $t = 3$	k = 11 $n = 16$ $t = 1$	k = 15 $n = 16$ $t = 0$	k = 16 $n = 16$ $t = 0$

Код Рида-Маллера Свойства кода **└** Параметры Возможные варианты



- 1. У красных кодов минимальное расстояние d равно единице они совершенно бесполезны, там количество кодов равно количеству сообщений; у желтых кодов d=2 они могут определить наличие ошибки, но не могут её исправить. Для всех остальных кодов d=2(t+1).
- 2. Напоминание: k длина сообщения, n длина кода, а t количество ошибок, которое код точно сможет исправить. Заодно о параметрах кода: m количество переменных у функции (очень влияет на длину кода), а r максимальная степень многочлена (очень влияет на длину сообщения, и соотвественно надёжность кода), причём r < m. Конечно, таблицу можно
- продолжать и дальше. 3. И кстати, случай m=0 , k=0 (не влез) будет собой представлять колирование единственного бита совершенно без изменений.

Как линейный код

Этот код является линейным кодом, к нему применимы все обычные (и неэффективные методы):

- Перебор по всему пространству кодовых слов в поисках ближайшего.
- lacktriangle С использованием синдромов: $s=rH^T$.

Код Рида-Маллера — Декодирование ∟Как линейный код

- 1. Этот способ применим ко всем кодам, но никто в здравом уме им не пользуется. 2. Здесь s синдром, r полученное сообщение, H проверочная матрица. Этот метод обычен для линейных кодов.
- 3. Эти способы нужно иметь в виду, но о них было рассказано и без меня, так что я их пропущу.

Определения

- \blacksquare Пусть $A\subseteq\{1,...,m\}$ для $m\in\mathbb{N}$
 - Подпространство $V_A\subseteq \mathbb{F}_2^m$, которое обнуляет все v_i , если $i\notin A$: $V_A=\{v\in \mathbb{F}_2^m: v_i=0\ \forall i\notin A\}$
 - \blacksquare Аналогично для $V_{\bar{A}}$, где $\bar{A}=\{1,...,m\}\setminus A\colon V_{\bar{A}}=\{v\in\mathbb{F}_2^m:v_i=0\ \forall i\in A\}$ Пример:
 - \blacksquare Пусть $m=3, A=\{1,2\}$, тогда...
 - $\blacksquare \ \mathbb{F}_2^m = \{ \texttt{000}, \texttt{001}, \texttt{010}, \texttt{011}, \texttt{100}, \texttt{101}, \texttt{110}, \texttt{111} \}$
 - $\ \ \, \mathbf{V}_{A} = \{ \mathbf{000}, \mathbf{010}, \mathbf{100}, \mathbf{110} \} \; (v_{3} = 0 \, \forall v) \;$
 - $\bar{A} = \{1, 2, 3\} \setminus A = \{3\}$
 - $\blacksquare \ V_{\bar{A}} = \{ \texttt{000}, \texttt{001} \} \ (v_1 = v_2 = 0 \, \forall v)$

Код Рида-Маллера — Декодирование **—**Алгоритм Рида —Определения

- 1. Начать стоит с нескольких определений, без которых алгоритм Рида объяснить не получится.
- 2. все 8 векторов этого пространства
 3. обнулилась третья позиция, первые две остались
- 4. осталась только третья позиция, остальные обнулились.



Смежные классы

Если фиксировано $V_A\subseteq \mathbb{F}_2^m$, то для каждого $b\in \mathbb{F}_2^m$ существует смежный класс

$$(V_A+b)=\{v+b\mid v\in V_A\}$$

Утверждается, что если брать $b \in V_{\bar{A}}$, то полученные смежные классы будут все различны (и это будут все смежные классы).

Код Рида-Маллера —Декодирование —Алгоритм Рида —Смежные классы

1. Почему все смежные классы (V_A+b) можно получить именно перебором $b\in V_{\bar A}$ можно найти в разделе «Дополнительные доказательства» из пдфки

Алгоритм Рида для кода $\mathrm{RM}(r,m)$

Декодирует сообщение u, если использовался $\mathrm{RM}(r,m)$. Для $\mathrm{RM}(2,2)$: $f(x_1,x_2)=u_{\{1,2\}}x_1x_2+u_{\{2\}}x_2+u_{\{1\}}x_1+u_{\varnothing}.$ Data: vector $y = (y_z \in \mathbb{F}_2 \mid z \in \mathbb{F}_2^m)$ for $t \leftarrow r$ to 0

c = 0 $\text{for each } b \in V_{\bar{A}}$ $c \mathrel{+=} \left(\sum_{z \in (V_A + b)} y_z\right) \bmod 2$ $u_A \leftarrow \mathbf{1} \left[c \geq 2^{m-t-1}\right]$ $y = \operatorname{Eval}\left(\sum_{A\subseteq \{1,\dots,m\}} u_A \prod_{i\in A} x_i
ight)$

На вход поступает бинарный вектор y длины $2^m.$ Это вектор значений функции, возможно с ошибками (но их не больше, чем $t=2^{m-r-1}-1$).

Код Рида-Маллера —Декодирование □Алгоритм Рида igsqcup Алгоритм Рида для кода $\mathrm{RM}(r,m)$

1. Теперь, наконец, сам алгоритм Рида с объяснением, что тут происходит. Почему он именно такой и почему это работает — см. раздел (на русском) «Reed's Algorithm: Unique decoding up to half the code distance» [??] в пдфке.

2. Цель — восстановить все коэффициенты при многочлене вида $f(x_1,...,x_m)=u_{\varnothing}+u_1x_1+x_2x_2+...+u_{1,2,...,r},$ где $\deg f\leq r$. Обратите внимание, что для индексов при u используются подмножества $A\subseteq\{1,...,m\},|A|\leq r$, причём каждый u_A умножается на моном $\prod_{i \in A} x_i.$

Алгоритм Рида для кода $\mathrm{RM}(r,m)$

 $f(x_1,x_2)=u_{\{1,2\}}x_1x_2+u_{\{2\}}x_2+u_{\{1\}}x_1+u_{\varnothing}.$ $\textbf{Data:} \ \mathrm{vector} \ y = (y_z \in \mathbb{F}_2 \mid z \in \mathbb{F}_2^m)$ for $t \leftarrow r$ to 0

 $\textbf{foreach}\ A\subseteq\{1,...,m\}\ \textit{with}\ |A|=t$ c = 0 $c += \left(\sum_{z \in (V_A + b)} y_z\right) \mod 2$ $u_A \leftarrow \mathbf{1} \left[c \geq 2^{m-t-1} \right]$ $y = \operatorname{Eval}\left(\sum_{\substack{A\subseteq\{1,\dots,m\}\\ |A|=t}} u_A \prod_{i\in A} x_i
ight)$

Декодирует сообщение u, если использовался $\mathrm{RM}(r,m)$. Для $\mathrm{RM}(2,2)$:

Будем восстанавливать сначала коэффициенты u_A при старших степенях, потом поменьше и так пока не восстановим их все. Hачинаем с t=r.

Код Рида-Маллера — Декодирование —Алгоритм Рида lacksquare Алгоритм Рида для кода $\mathrm{RM}(r,m)$

1. Теперь, наконец, сам алгоритм Рида с объяснением, что тут происходит. Почему он именно такой и почему это работает — см. раздел (на русском) «Reed's Algorithm: Unique decoding up to half the code distance» [??] в пдфке.

Алгоритм Рида для кода $\mathrm{RM}(r,m)$

Декодирует сообщение u, если использовался $\mathrm{RM}(r,m)$. Для $\mathrm{RM}(2,2)$: $f(x_1,x_2)=u_{\{1,2\}}x_1x_2+u_{\{2\}}x_2+u_{\{1\}}x_1+u_{\varnothing}.$

 $\textbf{Data:} \ \mathrm{vector} \ y = (y_z \in \mathbb{F}_2 \mid z \in \mathbb{F}_2^m)$ for $t \leftarrow r$ to 0

> $\textbf{foreach}\ A\subseteq\{1,...,m\}\ \textit{with}\ |A|=t$ c = 0

Хотим восстановить все коэффициенты при мономах степени t. Для этого перебираем все A, |A| = t и для каждого восстанавливаем коэффициент \boldsymbol{u}_A при $x_{A_{1}}x_{A_{2}}...x_{A_{t}}.$

Код Рида-Маллера — Декодирование — Алгоритм Рида lacktriangle Алгоритм Рида для кода $\mathrm{RM}(r,m)$

1. Теперь, наконец, сам алгоритм Рида с объяснением, что тут происходит. Почему он именно такой и почему это работает — см. раздел (на русском) «Reed's Algorithm: Unique decoding up to half the code distance» [??] в пдфке.

Алгоритм Рида для кода $\mathrm{RM}(r,m)$

for $t \leftarrow r$ to 0

Декодирует сообщение u, если использовался $\mathrm{RM}(r,m)$. Для $\mathrm{RM}(2,2)$: $f(x_1,x_2)=u_{\{1,2\}}x_1x_2+u_{\{2\}}x_2+u_{\{1\}}x_1+u_{\varnothing}.$ Data: vector $\boldsymbol{y} = (\boldsymbol{y}_z \in \mathbb{F}_2 \mid \boldsymbol{z} \in \mathbb{F}_2^m)$

Чтобы восстановить коэффициент, нужно перебрать

все смежные классы вида
$$(V_A+b)\colon$$

$$V_A=\{v\in\mathbb{F}_2^m \\ : v_i=0\ \forall i\notin A\}$$

$$b\in\{v\in\mathbb{F}_2^m \\ : v_i=0\ \forall i\in A\}$$

Код Рида-Маллера —Декодирование ∟Алгоритм Рида igsqcup Алгоритм Рида для кода $\mathrm{RM}(r,m)$

1. Теперь, наконец, сам алгоритм Рида с объяснением, что тут происходит. Почему он именно такой и почему это работает — см. раздел (на русском) «Reed's Algorithm: Unique decoding up to half the code distance» [??] в пдфке.

Код Рида-Маллера —Декодирование — _{Алгоритм} Рида lacksquare Алгоритм Рида для кода $\mathrm{RM}(r,m)$

- 1. Теперь, наконец, сам алгоритм Рида с объяснением, что тут происходит. Почему он именно такой и почему это работает см. раздел (на русском) «Reed's Algorithm: Unique decoding up to half the code distance» [??] в пдфке. 2. Если это количество больше порогового значения, то считаем, что $u_A=1$, иначе же $u_A=0$.

Код Рида-Маллера —Декодирование —Алгоритм Рида lacktriangle Алгоритм Рида для кода $\mathrm{RM}(r,m)$



1. Теперь, наконец, сам алгоритм Рида с объяснением, что тут происходит. Почему он именно такой и почему это работает — см. раздел (на русском) «Reed's Algorithm: Unique decoding up to half the code distance» [??] в пдфке.

Алгоритм Рида для кода $\mathrm{RM}(r,m)$

Декодирует сообщение u, если использовался $\mathrm{RM}(r,m)$. Для $\mathrm{RM}(2,2)$: $f(x_1,x_2) = u_{\{1,2\}} x_1 x_2 + u_{\{2\}} x_2 + u_{\{1\}} x_1 + u_{\varnothing}.$ $\textbf{Data:} \ \mathrm{vector} \ y = (y_z \in \mathbb{F}_2 \mid z \in \mathbb{F}_2^m)$ for $t \leftarrow r$ to 0

$$\begin{aligned} & \text{ for } t \leftarrow r \text{ to } 0 \\ & & \text{ foreach } A \subseteq \{1,...,m\} \text{ with } |A| = t \\ & c = 0 \\ & & \text{ foreach } b \in V_{\bar{A}} \\ & & & \left| \begin{array}{c} c + = \left(\sum\limits_{z \in (V_A + b)} y_z\right) \operatorname{mod} 2 \\ u_A \leftarrow 1 \left[c \geq 2^{m-t-1}\right] \\ y - & = \operatorname{Eval}\left(\sum\limits_{A \subseteq \{1,...,m\}} u_A \prod_{i \in A} x_i\right) \end{aligned}$$

Считаем количество (c)смежных классов, в которых $\sum_{z \in (V_A + b)} y_z = 1 \pmod{2}.$ Пороговое значение (2^{m-t-1}) здесь — половина от числа смежных классов. Таким образом, если большинство сумм дало 1, то $u_A=1$, иначе $u_A = 0$

Алгоритм Рида для кода $\mathrm{RM}(r,m)$

Декодирует сообщение u, если использовался $\mathrm{RM}(r,m)$. Для $\mathrm{RM}(2,2)$: $f(x_1,x_2)=u_{\{1,2\}}x_1x_2+u_{\{2\}}x_2+u_{\{1\}}x_1+u_{\varnothing}.$

 $\textbf{Data:} \ \mathrm{vector} \ y = (y_z \in \mathbb{F}_2 \mid z \in \mathbb{F}_2^m)$ for $t \leftarrow r$ to 0

 $\textbf{foreach}\ A\subseteq\{1,...,m\}\ \textit{with}\ |A|=t$ c = 0 $\text{for each } b \in V_{\bar{A}}$

Затем мы вычитаем из y(вектор значений функции) всё найденное на этой итерации, после чего переходим к мономам меньшей степени. Повторять до восстановления всех коэффициентов.

Пример

Ранее: 011 кодируется как 1100 при помощи ${
m RM}(1,2)$

$$\mathbf{101} \leadsto (f(x_1, x_2) = x_1 + 1) \leadsto \begin{pmatrix} x_1 & x_2 & f & y_{00} = 1 \\ 0 & 0 & 1 & y_{01} = 1 \\ 0 & 1 & 1 & y_{01} = 0 \\ 1 & 1 & 0 & y_{11} = 0 \end{pmatrix} \Longrightarrow \mathbf{1100}$$

Код Рида-Маллера —Декодирование ∟Алгоритм Рида **∟**Пример 1. Как происходит кодирование, схематически:

Пример

Ранее: 011 кодируется как 1100 при помощи ${
m RM}(1,2)$ Положим $y_{\theta\theta}=1$, $y_{\theta1}=1$, $y_{1\theta}=0$, $y_{11}=0$ Здесь m=2, значит $A\subseteq\{1,2\}$. Причём r=1, т.е. $|A|\le 1$.

Шаг 1/3: $t=1, A=\{1\}$

- \blacksquare Здесь $V_A = \{ \mathtt{00}, \mathtt{10} \}$, $V_{\bar{A}} = \{ \mathtt{00}, \mathtt{01} \}$. Нужно рассмотреть два смежных класса.
- $\qquad \qquad \bullet \ \, (V_A + \mathbf{00}) = \{\mathbf{00}, \mathbf{10}\} \text{, cymma: } y_{\mathbf{00}} + y_{\mathbf{10}} = 1 + 0 = 1$
- $\blacksquare \ (V_A + {\tt 01}) = \{ {\tt 01,11} \},$ сумма: $y_{\tt 01} + y_{\tt 11} = 1 + 0 = 1$
- \blacksquare Итого: $u_A=u_{\{1\}}=1$

Код Рида-Маллера —Декодирование —_{Алгоритм} Рида □Пример

- 1. Теперь начинаем декодирование. 2. (меняется только первый бит) 3. (первый бит обнулился) 4. по одному на каждый вектор из $V_{\tilde{A}}$

Пример

Ранее: 011 кодируется как 1100 при помощи $\mathrm{RM}(1,2)$ Положим $y_{00}=1,y_{01}=1,y_{10}=0,y_{11}=0$ Здесь m=2, значит $A\subseteq\{1,2\}$. Причём r=1, т.е. $|A|\le 1$.

Шаг 2/3: $t=1, A=\{2\}$

- lacktriangled Здесь $V_A=\{{\tt 00,01}\},\ V_{ar{A}}=\{{\tt 00,10}\}.$ Нужно рассмотреть два смежных класса
- $\blacksquare \ (V_A + \mathbf{00}) = \{\mathbf{00}, \mathbf{01}\}$, сумма: $y_{\mathbf{00}} + y_{\mathbf{01}} = 1 + 1 = 0$
- $\blacksquare \ (V_A + \mathbf{10}) = \{\mathbf{10}, \mathbf{11}\}, \text{ сумма: } y_{\mathbf{10}} + y_{\mathbf{11}} = 0 + 0 = 0$
- \blacksquare Итого: $u_A=u_{\{2\}}=0$

Код Рида-Маллера —Декодирование —Алгоритм Рида —Пример

1. — по одному на каждый вектор из $V_{ar{A}}$

Пример

Ранее: 011 кодируется как 1100 при помощи ${
m RM}(1,2)$ Положим $y_{00}=1,y_{01}=1,y_{10}=0,y_{11}=0$ Здесь m=2, значит $A\subseteq\{1,2\}$. Причём r=1, т.е. $|A|\le 1$.

Перед переходом к t=0, нужно вычесть из y вектор значений следующей функции:

$$g(x_1,x_2)=u_{\{2\}}x_2+u_{\{1\}}x_1=0x_2+1x_1=x_1$$

Вычислим $\mathrm{Eval}(g) \colon \underbrace{ \begin{array}{c|cc} x_1 & x_2 & g(x_1, x_2) \\ \hline 0 & 0 & 0 \end{array}}$ 0 1 0 1 0 1 1

Тогда $y \leftarrow y - \operatorname{Eval}(g) = \mathtt{1100} \oplus \mathtt{0011} = \mathtt{1111}.$

Код Рида-Маллера —Декодирование ∟Алгоритм Рида **∟**Пример



- 1. Здесь мы берём все u, полученные при t=1, домножаем каждую на соответствущие ей x-ы и получаем функцию от *m* переменных.
 2. Очень важно, чтобы у вас во всех таблицах истинности (в т.ч. той, которая использовалась при
- кодировании для получения y) был одинаковый порядок строх. Иначе чуда не выйдет. 3. Полезно заметить, что в \mathbb{F}_2 сложение и вычитание одно и то же.

Продолжение примера: t=0

Теперь
$$y_{\mathrm{\theta}\mathrm{0}} = 1, y_{\mathrm{\theta}\mathrm{1}} = 1, y_{\mathrm{10}} = 1, y_{\mathrm{11}} = 1$$

Шаг 3/3: $t=0, A=\emptyset$

- lacktriangle Здесь $V_A=\{ {
 m 00} \}$, но $V_{ar A}=\{ {
 m 00,01,10,11} \}.$ Нужно рассмотреть четыре смежных класса.
- $\blacksquare \ (V_A + \mathbf{00}) = \{\mathbf{00}\}$, сумма: $y_{\mathbf{00}} = 1$
- $\blacksquare \ (V_A + \mathbf{01}) = \{\mathbf{01}\}$, сумма: $y_{\mathbf{01}} = 1$
- $\blacksquare \ (V_A + \mathbf{10}) = \{\mathbf{10}\}$, сумма: $y_{\mathbf{10}} = 1$
- $\blacksquare \ (V_A + \mathbf{11}) = \{\mathbf{11}\} \text{, cymma: } y_{\mathbf{11}} = 1$
- Итого: $u_A = u_\varnothing = 1$

Продолжение примера: t=0

Теперь $y_{\mathsf{\theta}\mathsf{0}} = 1, y_{\mathsf{0}\mathsf{1}} = 1, y_{\mathsf{1}\mathsf{0}} = 1, y_{\mathsf{1}\mathsf{1}} = 1$

Получили $u_{\{2\}}=0, u_{\{1\}}=1, u_{\varnothing}=1.$

Это значит, что исходный многочлен был таков:

$$f(x_1,x_2)=u_{\{2\}}x_2+u_{\{1\}}x_1+u_\varnothing={\color{red}0}+x_1+{\color{black}1},$$

а исходное сообщение: 011, как и ожидалось.

Утверждается, что время работы алгоритма — $O(n\log^r n)$, где $n=2^m$ —



Домашнее задание

Вариант 1

- Закодировать сообщение: 1001.
- Декодировать кол. если ошибок нет: 1010, использовался RM(1, 2).
- 🖪 Декодировать код, полученный с ошибками: 1101 1010, использовался RM(1,3)

Вариант 2

- Закодировать сообщение: 0101.
- **2** Декодировать код, если ошибок нет: 0110, использовался ${\rm RM}(1,2).$
- 🖪 Декодировать код, полученный с ошибками: 1111 0100, использовался

Код Рида-Маллера —Домашнее задание

2022-03-

- 1. Замечание: каких-либо требований на методы решения нет, но если используете код приложите его. Различных способов решить существует больше одного. Номер варианта можете определять как $1+((5n+98) \bmod 2)$, но главное напишите его и своё
 - лима. Для кодирования использовался тот же порядок строк в таблице истинности, что и в остальной презентации; аргументы идут по столбцам слева направо по возрастанию номера. При формировании сообщения, слагаемые сортируются лексиографически, а затем по убыванию степени (см. примеры в презентации).



Источники

- 1 https://arxiv.org/pdf/2002.03317.pdf великолепный обзор, очень рекомендую.
- 2 http://dha.spb.ru/PDF/ReedMullerExamples.pdf очень хорошо и подробно, но используется подход через матрицы, а не через полиномы, а это не весело.
- 🖪 https://en.wikipedia.org/wiki/Reed-Muller_code кратко, чётко, понятно, но не описано декодирование.
- 4 https://ru.bmstu.wiki/Коды_Рида-Маллера в целом всё есть, но написано очень непонятно;