Информатика

Лекция 11

План лекции

- Алфавит, кодирование, код
- Типы кодирования, однозначное декодирование
- Метод кодирования Хафмана
- Метод кодирования Фано
- Элементы теорий вероятностей и информации – лекция 15
  - Модель информационной системы Шеннона
  - Среднестатистическая информационная емкость сообщений для эргодических источников с заданным распределением частот символов
  - Формулы Шеннона и Хартли для удельной емкости на символ
  - Избыточность кодирования

Понятие кода

- Алфавитом называется конечное множество символов
- Сообщением алфавита А называется конечная последовательность символов алфавита А
- Множество всех сообщений алфавита А обозначается А*

Понятие кода

- Кодом называется отображение К : Алф1* —> Алф2*, согласованное с конкатенацией, т.е. удовлетворяющее равенству К(с1с2...cN) = К(с1) К(с2)... К(cN) для любого сообщения с1с2...cN из Алф1*

- Значение К(с1с2...cN) называется кодом сообщения с1с2...cN

- Код К : Алф1* —> {0,1}* называется двоичным кодом

Кодирование и декодирование

- Кодированием сообщения называется вычисление кода сообщения
- Декодированием (дешифровкой) сообщения называется вычисление его прообраза под действием кода
- Код К называется однозначно декодируемым, если существует обратная функция $К^{-1}$
- Если вычисление $К^{-1}$ требует большого количества времени, то говорят не о кодировании, а о шифровании

Пример 1

Алф1 = {a,b,c,d}
Алф2 = {0,1}

К(a) = 0, К(b) = 01, К(c) = 10, К(d) = 1

$К^{-1}$(01101010) = {addbba, bccc, …} – прообраз 01101010

Данный код не является однозначно декодируемым

Пример 2

Алф1 = {a,b,c,d}
Алф2 = {0,1}

К(a) = 0, К(b) = 10, К(c) = 110, К(d) = 111

Почему данный код является однозначно декодируемым?

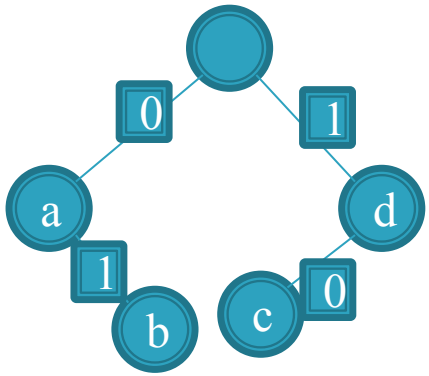
Кодовое дерево

Кодовым деревом кода К:Алф1 ->Алф2 называется такое дерево Т, с рёбрами помеченными символами из Алф2, что

- Любой путь из корня Т совпадает с началом кода какого-то символа из Алф1
- Код любого символа из Алф1 соответствует какому-то пути из корня Т
  - Почему не всегда до листа?

Пример кодового дерева

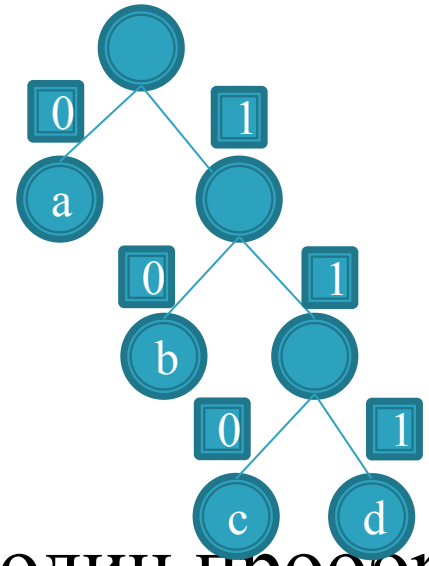
Алф1 = {a,b,c,d}
Алф2 = {0,1}
К(a) = 0, К(b) = 01,
К(c) = 10, К(d) = 1



Почему у сообщения 01101010 два прообраза?

Пример кодового дерева

Алф1 = {a,b,c,d}
Алф2 = {0,1}
К(a) = 0, К(b) = 10,
К(c) = 110, К(d) = 111



Почему у *любого* сообщения один прообраз?

Префиксный код

Код К называется префиксным, если для любых двух сообщений U и V код К(U) не является началом (префиксом) кода К(V) и наоборот

- Свойства префиксного кода
- В дереве префиксного кода коды всех символов заканчиваются в листьях

- Префиксный код позволяет выделять коды символов без использования разделителей

Примеры префиксных кодов

Пример 1

Алф1 = {a,b,c,d}
Алф2 = {0,1}

K(a) = 00, K(b) = 01, K(c) = 10, K(d) = 11

Как выглядит кодовое дерево этого кода?

Примеры префиксных кодов

Пример 2

Алф1 = {a,b,c,d}
Алф2 = {0,1}

К(a) = 0, К(b) = 10, К(c) = 110, К(d) = 111

Как выглядит кодовое дерево этого кода?

Однозначная декодируемость префиксного кода

Теорема Любой префиксный код однозначно декодируем

Доказательство

- Пусть К – префиксный код. Докажем, что у кода S=К(R) любого сообщения R ровно один прообраз
- Индукция по длине L сообщений R
- База L = 1
  - R восстанавливается однозначно в силу префиксности К
    - Что было бы, если бы коды *двух разных* символов являлись бы префиксом S
- Шаг L > 1
  - К согласован с конкатенацией ==> найдётся символ с такой, что S = К(c) S'
    - Что бы было бы, если бы такого символа не было бы или бы он был бы не один бы?
  - К префиксный ==> символ с единственный
  - Длина прообраза S' строго меньше длины прообраза S
  - По предположению индукции S' декодируется однозначно

Пример

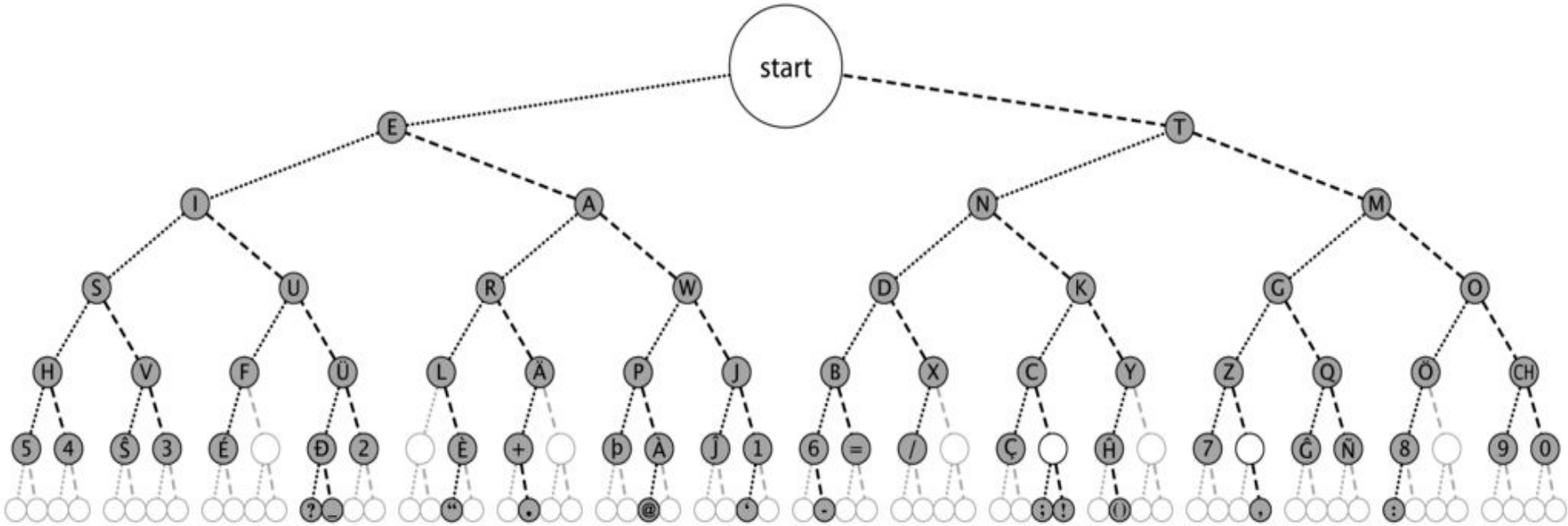
Алф1 = {a,b,c,d}
Алф2 = {0,1}

К(a) = 0, К(b) = 101, К(c) = 110, К(d) = 1110

Рассмотрим сообщение 01101010
01101010 = K(a) 1101010
1101010 = K(c) 1010
1010 = K(b) 0
0 = K(a)
K(acba) = 01101010

Пример азбука Морзе

- 1840 Alfred Vail по заказу телеграфной компании Samuel F.B. Morse
- Двоичный (точка, тире) непрефиксный код – почему?
- Троичный (точка, тире, пауза) префиксный код – почему?
- Кодовое дерево азбуки Морзе как двоичного кода для латиницы



Понятие оптимального кода

- Обозначим
  - Δ – множество кодов Алф1* -> Алф2*
  - К – какой-то код из Δ
  - R – произвольное сообщение из Алф1*
  - L(К, R) – длина R после кодирования
  - p_x – число вхождений символа c_x в R
    - заодно мы пронумеровали символы из Алф1, x – номер символа c_x
- Длина кода сообщения R есть $L(К,R) = \sum p_x \cdot L(К, c_x)$
- Код К* называется оптимальным для сообщения R в множестве кодов Δ, если
  $L(К^*,R) = \min \{ \text{длина}(К,R) \mid K \in \Delta \}$

Оптимальный двочиный префиксный код

- Как *быстро* построить оптимальный двоичный префиксный код для данного сообщения?

- Использование
  - Сжатие данных при хранении и передаче
  - Устранение избыточности при шифровании данных

- Алгоритм построения оптимального двоичного префиксного кода -- 1951, David A. Huffman, Massachusetts Institute of Technology
  - Оптимальный двоичный префиксный код не зависит от порядка символов в сообщении, только от частот отдельных символов
  - Связь с теорией информации

Свойства оптимального двоичного префиксного кода

Пусть R -- сообщение в алфавите Алф1={c1,…,cn}
c_x входит в R p_x раз (x=1,...,n)
K* -- оптимальный двоичный префиксный код для R

1. Если $p_x < p_y$, то $L_x(K^*) >= L_y(K^*)$
   - Иначе для кода $K(c_x) = K^*(c_y)$, $K(c_y) = K^*(c_x)$ и $K(c) = K^*(c)$ $L(K,R) < L(K^*,R)$

2. Можно занумеровать символы Алф1 так, чтобы $p_1>=p_2>=…>=p_n$ и $L(K^*,c_1)<=L(K^*,c_2)<=…<=L(K^*, c_n)$

Свойства оптимального двоичного префиксного кода

3. Символов с кодом длины $L(K^*, c_n)$ (с самым длинным кодом) не менее двух
   - Иначе удалим последний символ в коде c_n -- длина $L(K^*, R)$ сократится, префиксность K^* сохранится

4. Можно перенумеровать символы так, что $K^*(c_n) = P\ 0$ и $K^*(c_{n-1}) = P\ 1$ и сохранив условие 2
   - Следует из свойства 3

Свойства оптимального двоичного префиксного кода

5. Оптимальный двоичный префиксный код к* для сообщения r, полученного из сообщения R заменой самого редкого символа c_n на c_{n-1} , и К* связаны соотношениями
   - к*(c_{n-1}) = удалить из К*(c_{n-1}) последний символ
   - К*(c_n) = к*(c_{n-1}) 0
   - К*(c_{n-1}) = к*(c_{n-1}) 1
   - К*(c) = к*(c) для остальных символов c
   - L(K*,R) = L(k*,r) + p_n + p_{n-1}

Построение дерева оптимального префиксного двоичного кода

Вход

Кратности p1, …, pn вхождений симолов c1, ..., cn в сообщение

Выход

Дерево оптимального двоичного префиксного кода для сообщения

Алгоритм

- W = {p1(c1), …, pn(cn)} – множество деревьев
  - Левая скобочная запись, кратности в качестве меток вершин
- пока в W два или более поддеревьев
  - Найти в W деревья T = x(...) и U = y(...) с минимальными метками x и y
  - W = (W \ {T, U}) U { (x+y)(T, U) }

Пример

кол около колокола

о – 7; к – 4; л – 4; пробел – 2; а – 1.

Один из вариантов работы алгоритма

Множество W

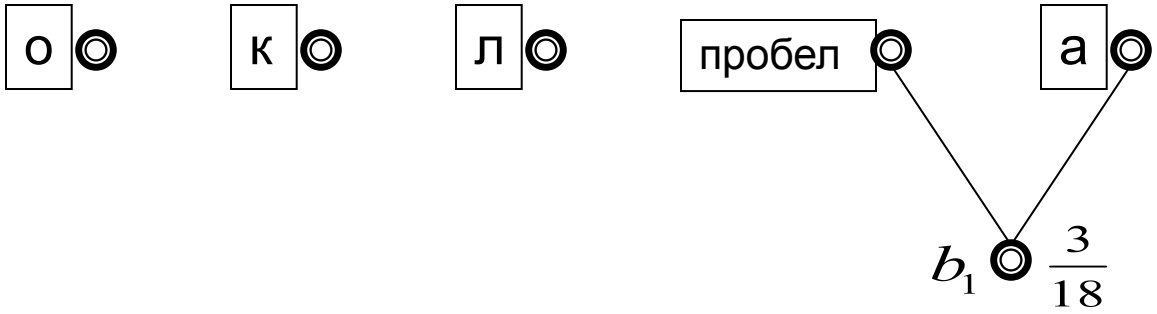
До цикла {7(о), 4(к), 4(л), 2(пробел), 1(а) }

После шага 1 {7(о), 4(к), 4(л), 3(2(пробел), 1(а)) }

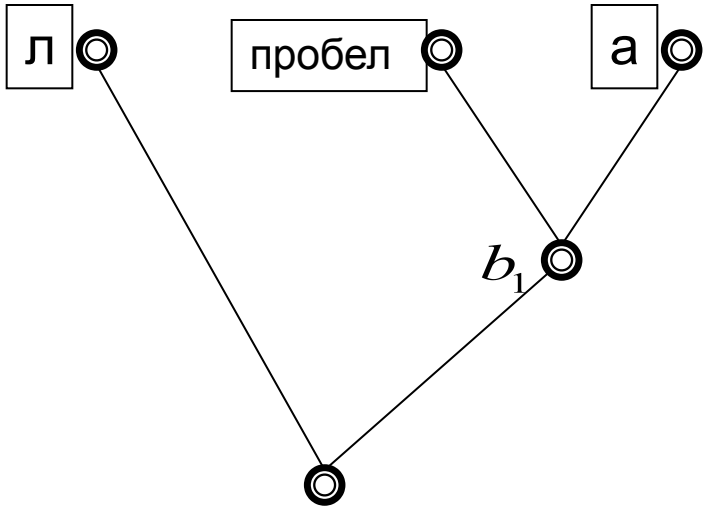
После шага 2 {7(о), 4(к), 7(4(л), 3(2(пробел), 1(а))) }

После шага 3 {7(о), 11(4(к), 7(4(л), 3(2(пробел), 1(а)))) }

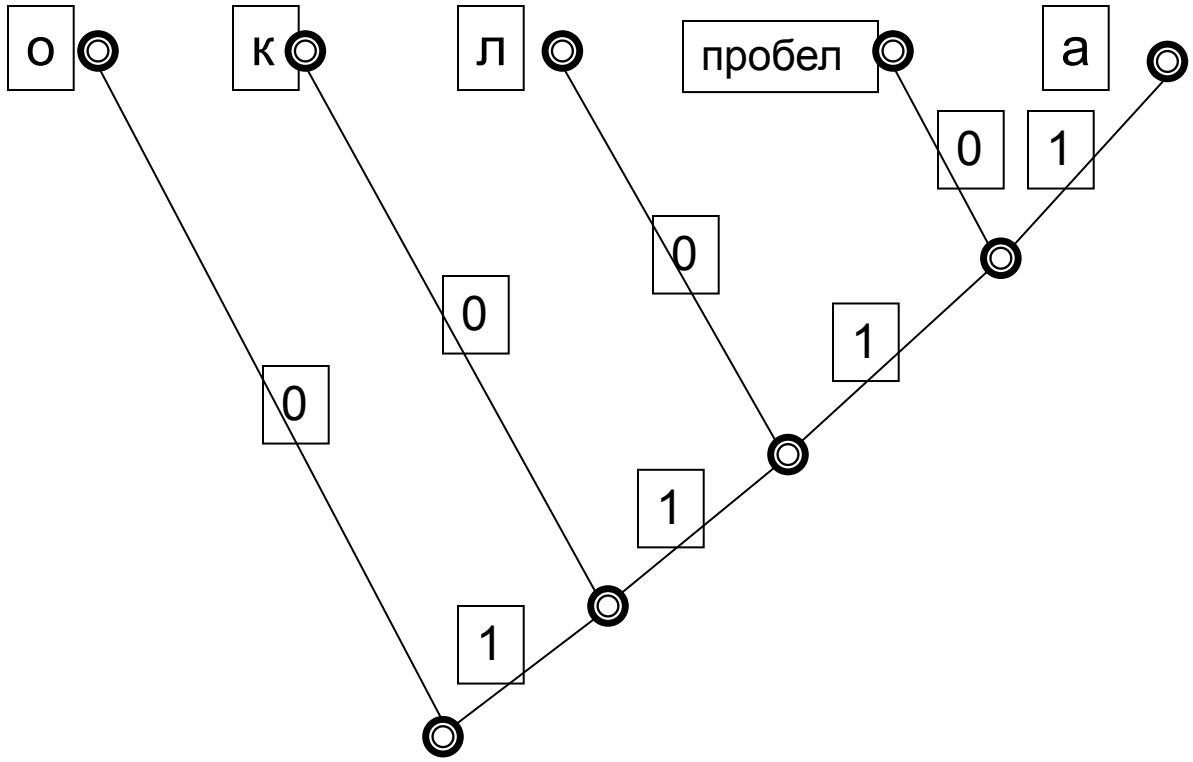
После шага 4 {18(7(о), 11(4(к), 7(4(л), 3(2(пробел), 1(а))))) }



Дерево после шага 1



Дерево после шага 2



Дерево после шага 4

Пример построения кода по кодовому дереву

- Пометим дуги, исходящие из каждой вершины дерева, единицей и нулем
- Проходя путь из корня дерева до символа и выписывая все пометки дуг на этом пути, получим код для этого символа

В нашем примере коды будут такими

о 0,

к 10 пробел 1110

л 110а 1111

Закодированное сообщение

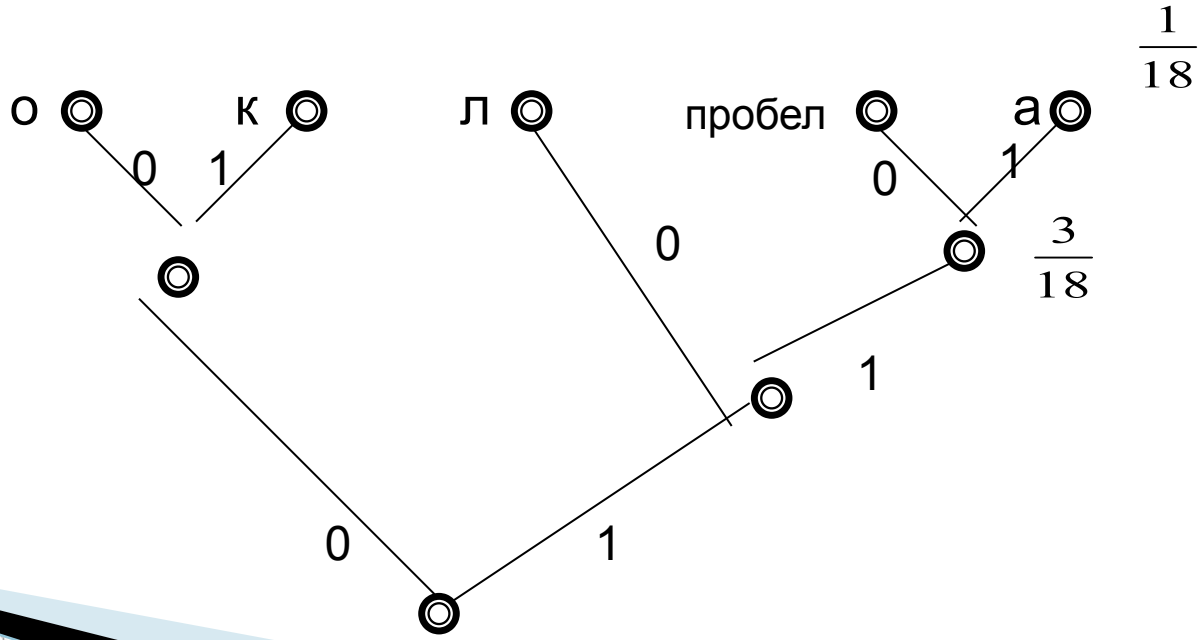
1001101110010011001110100100100110111 1

Длина закодированного сообщения L = 39

Для разобранного примера можно построить другое дерево

Закодированное сообщение длины L = 39

010010110000100100011001001000010010111



Теорема

Длина кодового слова в оптимальном префиксном двоичном коде ограничена порядковым номером минимального числа Фибоначчи, превосходящего длину входного текста.

Доказательство – в качестве упражнения

Следствие

При кодировании по алгоритму Хаффмана текстов ASCII размером до 11Тб код любого символа короче 64 битов

- Алфавитное кодирование. Код
- Типы кодирования, однозначное декодирование
- Метод кодирования Хафмана
- Метод кодирования Фано
- Элементы теорий вероятностей и информации – лекция 15
  - Модель информационной системы Шеннона
  - Среднестатистическая информационная емкость сообщений для эргодических источников с заданным распределением частот символов
  - Формулы Шеннона и Хартли для удельной емкости на символ
  - Избыточность кодирования

Метод Фано

Роберт Марио Фано р. 1917

Один из первых алгоритмов сжатия на основе префиксного кода

Метод Фано

- Упорядочим входной алфавит по возрастанию *частот* p1 <= p2 <= … <= pn вхождения символов в сообщение
- Обозначим Sk = p1+p2+…+pk, S0 = 0
- Строим таблицу K с двоичными кодами символов входного алфавита
- K[i][1] = i-й символ (по возрастанию частот)
- K[i][2] = Sk
- Остальные клетки – на след. слайде

Метод Фано

- K[i][j] заполняем 0 и 1 по след. правилу
- Для каждого *максимального* интервала строк [a, b], у которых в столбце j-1 находятся одинаковые цифры
  - Находим c $\in$ [a, b] такое, что Sc ближе всего к (Sa+Sb)/2
  - K[i][j] = 1 для i $\in$ [a, c], K[i][j] = 0 для i $\in$ [c+1, b]

Пример

A = {a, b, c, d, e}

Частоты $p_a = 0.11$, $p_b = 0.15$, $p_c = 0.20$, $p_d = 0.24$, $p_e = 0.30$

0.46 ближе к 0.5

0.26 ближе всех к (0.00+0.46)/2=0.23

0.70 ближе всех к (0.46+1.00)/2=0.73

0.11 ближе всех к (0.00+0.26)/2=0.13

	P_i	S_i			
		0			
a	0.11	0.11	1	1	1
b	0.15	0.26	1	1	0
c	0.20	0.46	1	0	
d	0.24	0.70	0	1	
e	0.30	1.00	0	0	

Свойства кода Фано

- Кодовое дерево для кода Фано обладает следующим свойством
  - Ребра, исходящие из корня, соответствуют разбиению алфавита на две группы символов, близкие по частоте
  - Ребра, исходящие из вершины следующего «этажа», соответствуют разбиению соответствующей группы на близкие по частоте подгруппы и т. д.
- Код Фано – префиксный код
  - Почему?

Свойства кода Фано

- Код Фано неоптимальный

- Пример
  - Частоты p1=0.4, p2=p3=p4=p5=0.15
  - Фано: 00 01 10 110 111
    - средняя длина кодового слова 2*0.4+(2+2)*0.15+(3+3)*0.15 = 2.3
  - Хаффман: 0 010 011 000 001
    - средняя длина кодового слова 1*0.4+ (3+3+3+3)*0.15 = 2.2
  - Как выглядят кодовые деревья кода Хаффмана т Фано?

Метод Шеннона

- Клод Шеннон 1916 – 2001, основоположник теории информации

1. Упорядочим входные символы по возрастанию частот и образуем частичные суммы Sk как в методе Фано
2. Для каждой частоты Sk находим nk т.ч. $1/2^{nk} \leq Sk \leq 2/2^{nk}$ --- нужно отделить одну Sk от другой
3. Sk разлагаем в двочную дробь $0.d_1d_2d_3\ldots$
4. Первые nk цифр этой дроби задают код для k-го символа

Пример построения кода Шеннона

		nk	разложение Sk	код
p(a) = 0.08	$S_a = 0.08$	4	0.0001	0001
p(b) = 0.12	$S_b = 0.20$	4	0.0011	0011
p(c) = 0.15	$S_c = 0.35$	3	0.010	010
p(d) = 0.28	$S_d = 0.63$	2	0.10	10
p(e) = 0.37	$S_d = 1.00$	2	0.11	11

Пример вычисления na:

0.08 ~= 1/12; $1/2^4 \leq 1/12 \leq 2/2^4$

Свойства кода Шеннона

- Код Шеннона -- префиксный код
  - Почему?

- Пусть pk – частота вхождения k-го символа в кодируемое сообщение длины N. Кодирование такого сообщения кодом Шеннона дает сообщение длины не более N*(p1*log2(p1) + p2*log2(p2) + … + pn*log2(pn))
  - Почему? Как Шеннон выбрал длины кодовых слов?

Элементы теории информации

Лекция 15

Информационная модель Клода Шеннона

- The Bell System Technical Journal Vol. 27, pp. 379–423, 623–656, July, October, 1948
- Имеются источник (кодер) и приемник (декодер)
- Они связаны между собой каналом передачи символ
  - Символы – пример дискретного сигнала
- Канал не искажает и не теряет символы

- Какой нужен канал, чтобы передать данное сообщение (последовательность символов) за данное время?
- За какое время можно передать данное сообщение по данному каналу?
- За какое время *нельзя* передать данное сообщение по данному каналу без потерь?

- Шеннон исследовал также передачу непрерывного сигнала и передачу с шумом

Информационная модель Клода Шеннона

- Каким должен быть канал, чтобы передать данное сообщение за данное время?
- За какое время можно передать данное сообщение по данному каналу?
- Как измерять пропускную способность канала?
  - Если передача всех символов занимает одинаковое время, то используем символы в секунду
  - Как быть, если передача разных символов занимает разное время?

Информационная модель Клода Шеннона

- Как измерять пропускную способность канала?
  - Если передача всех символов занимает одинаковое время, то можно использовать символы в секунду
  - Как быть, если передача разных символов занимает разное время?
- Пусть N(T) – число допустимых сообщений, передача которых занимает время T
- *Пропускная способность* = предел log2(N(T))/T при T --> oo
- Выбор log2 обусловлен математическим и интуитивным удобством
  - Если появляется возможность передавать за время T на один двоичный символ больше, то N(T) возрастает в два раза
  - Пропускная способность – на 1/T
  - Без скорость, вычисленная без log2, увеличилась бы в два раза

Информационная модель Клода Шеннона

- За какое время *нельзя* передать данное сообщение по данному каналу без потерь? Как понять, что источник порождает больше
- Как измерить скорость, с которой источник порождает информацию?
  - В общем случае – каково минимальное число 0 и 1, необходимых для однозначного восстановления сообщения с помощью подходящего алгоритма -- *алгоритмическая сложность* Коломогорова – алгоритмически невычислимая величина для произвольных сообщений

Информационная модель Клода Шеннона

- Как измерить скорость, с которой источник порождает информацию?

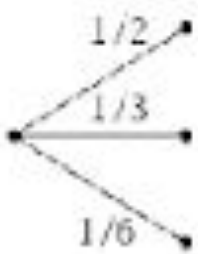
- В процессе передачи сообщения источник "помогает" приемнику выбрать один из символов
  - При условии наличия у приемника и источника общего знания о передаваемом сообщении
- Какое количество "выбора" содержится в каждом символе?

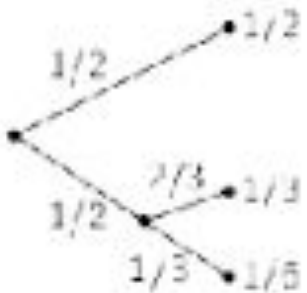
- Шеннон рассмотрел случай, когда известны только частоты отдельных символов p1, p2, …, pn

Информационная модель Клода Шеннона

- Для случая , когда приемник и передатчик знают только частоты отдельных символов p1, p2, …, pn, Шеннон сформулировал три требования к количеству "выбора" H(p1, p2, …, pn)

1. H должна быть непрерывна по pk
2. Значение H(1/n, 1/n, …, 1/n) должна возрастать по числу символов n
3. $H(p1, p2, \ldots, pn) = H(p1, \ldots, p_{n-1}+pn) + (p_{n-1}+pn)H(p_{n-1}/(p_{n-1}+pn), pn/(p_{n-1}+pn))$
   - H(1/2,1/3,1/6) = H(1/2,1/2)+1/2H(2/3,1/3)





Информационная модель Клода Шеннона

- Теорема Все функции, удовлетворяющие условиям 1-3, имеют вид

$$H = - c \sum pk \log(pk)$$

Будем говорить, что источник передал приемнику некоторую *информацию* о происшедшем событии, на основании которой изменилось представление приемника о множестве возможных исходов наблюдаемой величины.

Определим *количество информации*, содержащейся в сообщении *m*, изменяющем представление приемника о событии с $S_{ДО}$ до $S_{ПОСЛЕ}$ по формуле

(2)

$$I(m) = -\log_2 \frac{p(S_{до})}{p(S_{после})}$$

Единицей количества информации является *бит*.

Пример 1

В семье должен родиться ребенок.
Пространство элементарных исходов данной случайной величины — *{мальчик, девочка},* — состоит из двух исходов.
Отсутствие априорной информации у приемника (родителей) о поле малыша означает, что $S_{ДО}$ совпадает с этим пространством.
Сообщение источника (врача) «у вас родился мальчик» сужает это множество предположений до множества $S_{П0СЛЕ}$ из единственного исхода *мальчик.*
По формуле (12) количество полученной информации определяется как

$$I(m) = -\log_2 \frac{p(S_{ДО})}{p(S_{ПОСЛЕ})} = -\log_2 \frac{1}{2} = 1(\text{бит}).$$

$$log_2 2 = 1 \ - ?$$

- 1 бит соответствует сообщению о том, что произошло одно из двух равновероятных событий;
- требуется один бит для хранения сообщений о двух равновероятных событиях.

Пример 2

Из колоды вытягивается карта. Пространство элементарных исходов — 52 карты. В отсутствие изначальной информации пространство предположений $S_{ДО\_1}$ совпадает со всем пространством.
Первое сообщение от источника «выпала трефа» сужает его до $S_{ПОСЛЕ\_1}$ из 13 возможных исходов.
Второе сообщение «выпала картинка» сужает $S_{ДО\_2} = S_{ПОСЛЕ\_1}$ до $S_{ПОСЛЕ}$ состоящего из 4 исходов.
Третье сообщение «выпала дама треф» сужает $S_{ДО\_3} = S_{ПОСЛЕ\_3}$ до $S_{ПОСЛЕ\_3}$, состоящего из единственного исхода.
Количество информации, содержащееся в первом сообщении равно $-log_2\ 13/52 = 2$ битам, во втором — $-log_2\ 4/13 = 1.5$, в третьем — $-log_2\ 1/4 = 2$ битам.
Нетрудно проверить, что суммарное количество полученной информации — 5.5 бит, совпадает с количеством информации, которое несло бы сообщение «выпала дама треф» = $-log_2\ 1/52 = 5.5$ бит.

Теорема об аддитивности информации

Теорема

Количество информации, переносимое сообщением m1 && m2 && … && mN, не зависит от порядка отдельных сообщений и равно сумме количеств информации, переносимых сообщениями m1, …, mN по отдельности.

Выберем какой-либо порядок передачи сообщений

I(W, m1) = log2(P(m1)/P(W))

I(m1, m1&&m2) = log2(P(m1&&m2)/P(m1))

I(m1 && m2 && … && m_N-1, m1 && m2 && … && mN) = log2(P(m1&&…&&mN)/P(m1&&…m_N-1))

Пример о двух источниках:

1 – p(что грань 5)=1; $log\ P_{после}/P_{до} = log\ 1/1 = 0$;

2 – p(что грань 5)=1/6; $log\ P_{после}/P_{до} = log\ 1/1/6 = log\ 6 \approx 2{,}5$ бит.

Свойства информации:

— количество полученной приемником информации зависит от его предварительного знания о событии;

Формулы Шеннона, Хартли

Предположим теперь, что источник является генератором символов из некоторого множества $\{x_1, x_2, ..., x_n\}$ (назовем его алфавитом источника). Эти символы могут служить для обозначения каких-то элементарных событий, происходящих в области источника, но, абстрагируясь от них, в дальнейшем будем считать, что рассматриваемым событием является поступление в канал самих символов.

Если $p(x_i)$ — вероятность поступления в канал символа x_i, то

$$\sum_{i=1}^{n} p(x_i) = 1.$$

Рассмотрим теперь модель, в которой элементарным исходом является текстовое *сообщение*. Таким образом, Ω — это множество всех цепочек символов произвольной длины.

По поступившему сообщению m можно посчитать экспериментальную *частоту* встречаемости в нем каждого символа, где N — общая длина сообщения, а n_i — число повторений в нем символа x_i.

$$\nu_m(x_i) = \frac{n_i}{N},$$

Понятно, что анализируя различные сообщения, мы будем получать различные экспериментальные частоты символов, но для источников, характеризующихся закономерностью выдачи символов (их называют эргодическими), оказывается, что в достаточно длинных сообщениях все частоты символов сходятся к некоторым устойчивым величинам которые можно рассматривать как *распределение вероятностей* выдачи символов данным источником.

(4)

$$p(x_i) = \lim_{N \to \infty} \frac{n_i}{N},$$

Рассмотрим сообщение m, состоящее из n_1 символов x_1, n_2 символов x_2 и т. д. в произвольном порядке, как серию элементарных событий, состоящих в выдаче одиночных символов.

Тогда вероятность появления на выходе источника сообщения m равна

$$p(m) = \frac{(n_1)^{n_1}}{N} \cdot \ldots \cdot \frac{(n_n)^{n_n}}{N} = \frac{1}{N^N} \cdot (n_1^{n_1} \cdot \ldots \cdot n_n^{n_n}).$$

Количество информации, переносимой сообщением m длины N, определяется как

$$I(m) = -\log_2 \frac{p(m)}{1} = -\log_2 \left(\left(\frac{n_1}{N} \right)^{n_1} \cdot \ldots \cdot \left(\frac{n_n}{N} \right)^{n_n} \right) = -\sum_{i=1}^{N} n_i \cdot \log_2 \left(\frac{n_i}{N} \right).$$

Количество информации, приходящейся в среднем на каждый символ в сообщении m, есть

$$I_0(m) = \frac{1}{N} \cdot I(m),$$

где N — длина сообщения m.

Формула **Шеннона**

Перейдем к пределу по длине всевозможных сообщений *(N —> ∞)*:

$$I_0(A) = \lim_{N\to\infty} I_0(m) = \lim_{N\to\infty} \frac{1}{N} \cdot \left(-\sum_{i=1}^{N} n_i \cdot \log_2 \left(\frac{n_i}{N} \right) \right) =$$

$$= \left(-\sum_{i=1}^{N} \lim_{N\to\infty} \left(\frac{n_i}{N} \right) \cdot \log_2 \lim_{N\to\infty} \left(\frac{n_i}{N} \right) \right).$$

По формуле (14), вспоминая, что в достаточно большом сообщении $p(x_i) = \lim N\text{->}\infty$, получаем

$$\frac{n_i}{N}$$

$$I_0(A) = -\sum_{i=1}^{N} p(x_i) \cdot \log_2 p(x_i). \qquad (5)$$

Формула Хартли

Величина $I_0\ (A)$ характеризует среднее количество информации на один символ из алфавита A *с* заданным (или экспериментально определенным) распределением вероятностей

$$p(x_1),\ p(x_2),\ \ldots\ ,\ p(x_N).$$

Рассмотрим случай, когда все символы в алфавите равновероятны:

$$p(x1)\ = p(x2)\ \ \ldots\ \ = p(x_N) = \ 1/N\ \ .$$

Среднее количество информации, приходящееся на каждый символ такого алфавита, по формуле Шеннона

$$I_0(A) = -\sum_{i=1}^{N} \frac{1}{N} \cdot \log_2\left(\frac{1}{N}\right) = -N \cdot \frac{1}{N} \cdot \log_2 \frac{1}{N} = \log_2 N. \qquad (6)$$

Событие, которое может произойти или нет, называют *случайным*.

Примеры: попадание стрелка в мишень,
извлечение дамы пик из колоды карт,
выигрыш билета в розыгрыше лотереи и т. д.

На основании отдельно взятого случайного события нельзя научно предсказать, например, какие билеты окажутся выигрышными. Но если провести достаточно большую последовательность испытаний, то можно выявить определенные закономерности, позволяющие делать количественные предсказания.

Определение

Пространство элементарных событий (исходов) Ω – множество всех различных событий, возможных при проведении эксперимента.

Элементарность исходов понимается в том смысле, что ни один из них не рассматривается как сочетание других событий.

Примеры:

1) Будем бросать монету до тех пор, пока не выпадет герб. После этого эксперимент закончим.
«Элементарный исход» этого эксперимента можно представить в виде последовательности
р, р, р, ..., р, г (где р — решка, г — герб).
Таких последовательностей бесконечно много.
Следовательно, в данном случае множество Ω бесконечно.
2) Однократное бросание игральной кости. Будем считать, что возможен только один из 6 исходов, соответствующих падению кости гранями с 1, 2,...,6 очками вверх. Каждый возможный исход удобно обозначать числом выпавших очков.
Тогда пространство элементарных событий $\Omega = \{1,2,3,4,5,6\}$.

Формула $\omega \in \Omega$ означает, что элементарное событие ω является элементом пространства Ω.
Многие события естественно описывать множествами, составленными из элементарных исходов.
Например, событие, состоящее в появлении четного числа очков, описывается множеством $S = \{2,4,6\}$.
Формула $S \subseteq \Omega$ означает, что событие S является подмножеством пространства Ω.

- *Случайная величина —> переменная*
- *Элементарный исход —> значение переменной*
- *Пространство элементарных исходов —> область значений*
- *Событие —> подмножество области значений*

Определим формально *меру события μ*, как отображение из пространства Ω в *N*, обладающее следующими свойствами:

1) $\mu(\theta) = 0,$ где θ - пустое множество, т.е. множество, не содержащее ни одного элемента;

2) $S_1 \subseteq S_2 \Rightarrow \mu(S_1) \le \mu(S_2), \quad S_1 \subseteq \Omega, S_2 \subseteq \Omega;$

3) mu(S1 U S2)=mu(S1)+mu(S2)-mu(S1/\S2)

$\mu(S_1 \cup S_2) = \mu(S_1) + \mu(S_2) - \mu(S_1 \cap S_2).$

Введем функцию *p(S) вероятности события* как численного выражения возможности события *S* на заданном пространстве элементарных исходов Ω следующим образом:

$$p(S) = \frac{\mu(S)}{\mu(\Omega)} = \frac{\text{Число желательных исходов}}{\text{Число всех возможных исходов}} \quad (1)$$

«Желательные» исходы - элементарные исходы, образующие событие S.

$0 \le p(S) \le 1$ $p(0) = 0$, $p(\Omega) = 1$.

Событие с вероятностью 1 содержит все элементарные исходы и,
следовательно, происходит наверняка.

Событие с вероятностью 0 не содержит ни одного исхода, следовательно, не происходит никогда.

Говорят, что заданы вероятности элементарных событий, если на Ω задана неотрицательная числовая функция p такая, что:

$$\sum_{\omega \in \Omega} p(w) = 1.$$

Вероятность того, что при бросании кости выпадет единица,

равна $$\frac{\mu(\{1\})}{\mu(\{1,2,3,4,5,6\})} = \frac{1}{6}.$$

Вероятность появления четного числа очков равна

$$\frac{\mu(\{2,4,6\})}{\mu(\{1,2,3,4,5,6\})} = \frac{3}{6} = \frac{1}{2}.$$

Паскаль в письмах к Ферма в 1654 г. писал:
«Как велика вероятность, что когда я проснусь ночью и посмотрю на часы, то большая стрелка будет стоять между 15 и 20 минутами?»
И в этом же письме приводит рассуждения о том, что вероятность того, что стрелка часов будет находиться в этом промежутке, равна 5/60=1/12.

Теорема о сложении вероятностей

Если пересечение событий *A* и *B* непусто, то

$$p(A \cup B) = p(A) + p(B) - p(A \cap B).$$

(Это следует из аксиомы 3 для меры.)

Пример. Найдем вероятность того, что вытащенная из полной колоды карта окажется пикой или картинкой.

Пусть событию *A* соответствует извлечение из колоды карт пики, событию *B* — картинки.

Для каждой карты из колоды вероятность вытащить ее равна 1/52. Число пик в полной колоде равно 13. Следовательно, вероятность события *A* равна 13/52=1/4. Число картинок равно 16, вероятность события *B* равна 16/52 = 4/13.

События *A* и *B* имеют непустое пересечение. Множество $A \cap B$ состоит из четырех элементов,следовательно, $p(A \cap B) = 4/52 = 1/13$.

$$p(A \cup B) = p(A) + p(B) - p(A \cap B = 1/4+4/13-1/13=25/52.$$

Вероятность того, что вытащенная из полной колоды карта окажется пикой или червой равна равна 1/4 + 1/4 = 1/2.

Теорема об умножении вероятностей

Рассмотрим теперь серию экспериментов, в которой некоторая случайная величина наблюдается последовательно несколько раз. Последовательные события называются *независимыми*, если наступление каждого из них не связано ни с каким из других.

Например, исходы при бросании кости являются независимыми событиями, а последовательные вытягивания карт из одной и той же колоды без возврата — нет.

Теорема. Вероятность того, что независимые события S1, S2 произойдут в одной серии испытаний, равна произведению вероятностей событий *S*1 и *S*2.

Вероятность того, что обе монеты упадут гербом вверх равна 1/2 * 1/ 2 = 1/4.

Определим формально *меру события* μ, как отображение из пространства Ω в N, обладающее следующими свойствами:

1) $\mu(\emptyset) = 0$, где $\emptyset$ — пустое множество, т. е. множество, не содержащее ни одного элемента;

2) $S_1 \subseteq S_2 \Rightarrow \mu(S_1) \leq \mu(S_2)$, где $S_1 \subseteq \Omega, S_2 \subseteq \Omega$;

3) $\mu(S_1 \cup S_2) = \mu(S_1) + \mu(S_2) - \mu(S_1 \cap S_2)$.

КОНЕЦ ЛЕКЦИИ

Избыточность кодирования

Оказывается, что величина $I_0(A)$ определяет предел сжимаемости кода: никакой двоичный код не может иметь среднюю длину меньшую, чем I_0, в противном случае можно было бы передать некоторое количество информации меньшим числом битов, что невозможно.

Таким образом, любой код может быть лишь в большей или меньшей степени *избыточным*.

Относительная избыточность кода характеризуется как отношение числа «избыточных» битов в коде к общей длине кода,

то избыточное число битов есть $L - N * I_0(A)$,

(сообщение из N символов алфавита A с информационной емкостью $I_0(A)$, код длины L битов) а удельная избыточность каждого символа кода:

$$\frac{L - N \cdot I_0(A)}{L} = 1 - \frac{N}{L} \cdot I_0(A). \qquad (7)$$

Заметив, что $\lim_{N->\infty}$ L/N - есть средняя длина кодового слова $K_0(A)$, получим независимое от сообщения соотношение для избыточности кода:

$$Z(K) = 1 - I_0(A)/K_0(A).$$

Оптимальный код с нулевой избыточностью является код со средней длиной кодового слова $K_0 = I_0(A)$ битов или наиболее близкий к нему.

Резюме. $I_0(A)$ показывает, какое в среднем количество двоичных символов нужно для записи всех кодовых слов алфавита А при произвольном кодировании «символ —> слово».

Для алфавитов с равновероятными символами формула Хартли определяет минимальную необходимую длину кодового слова, например для алфавита *ASCII*: $I_0(ASCII) = log_2 256 = 8$ бит.

Таким образом, любой 8-битный код для *ASCII* будет оптимальным.

Посчитаем информационную емкость кода: длина исходного сообщения $N = 18$, длина кода $L = 39$ битов.

Удельная информационная емкость алфавита A с распределением

P есть

$$I_0(A) = \frac{8}{18}\cdot\log_2\frac{18}{4} + \frac{1}{18}\cdot\log_2\frac{18}{1} + \frac{7}{18}\cdot\log_2\frac{18}{7} + \frac{2}{18}\cdot\log_2\frac{18}{2} = 2.1 \quad .$$

Избыточность кода

$$Z = 1 - \frac{N}{L}\cdot I_0(A) = 1 - \frac{18}{39}\cdot 2.1 = 0.03,$$

Реализация проекта

Архиватор должен вызываться из командной строки,

формат вызова:

`harc.exe -[axdlt] arc[.ext] file_1 file_2 … file_n`

Поддерживаемые операции:

- *a*- поместить файл(ы) в архив;
- *x* - извлечь файл(ы) из архива;
- *d* - удалить файл(ы) из архива;
- *l* - вывести информацию о файлах, хранящихся в архиве;
- *t* - проверить целостность архива.

Проверка целостности архива

```
_stat, _wstat, _stati64, _wstati64
int _stat(const char* path, struct _stat *buffer);

#include <sys/stat.h>
```

CRC32 – проверка контрольных сумм

Построение дерева Хаффмана

Вход:

A – исходный набор символов $<a_1,...,a_N>$,

$P=<p_1,p_2,...,p_N>$ - распределение их частот;

– $W_0 = \{<a_1,p_1>,...,<a_N,p_N>\}$ (начальный набор свободных узлов соответствует встречающимся символам);

– цикл по i от 0 до N-1

W_i = Шаг_построения(W_{i-1});

Выход:

Дерево Хаффмана, построенное в цикле с корневым узлом, содержащимся в W_N.

Код Хаффмана

Алгоритм:

1. Определить алфавит $A = \{ c_1, c_2, \ldots, c_n \}$ сообщения S и подсчитать число вхождений $p_1, p_2, \ldots, p_n$ в S
2. Построить дерево оптимального префиксного двоичного кода для S используя свойства 1-8 оптимального кода – полученный префиксный двоичный код называется кодом Хаффмана (1951, David A. Huffman, Massachusetts Institute of Technology)
3. Закодировать сообщение S используя код Хаффмана

Критерии качества кодирования:

— минимальная длина кода;
— однозначное декодирование.

Информационная модель Клода Шеннона

- Пусть в области источника происходит наблюдение за некоторой случайной величиной.
- Приемник может иметь некоторое априорное представление о множестве $S_{до}$ возможных исходов этой величины до того, как произошло наблюдение.
- Когда ничего не известно заранее, $S_{до}$ принимается за все пространство возможных исходов Ω.
- Источник передает приемнику сообщение о произошедшем наблюдении, после получения которого множество предположительных исходов у приемника сужается до $S_{ПОСЛЕ}$.
- Это представление будем называть апостериорным.