Уральский федеральный университет

ИНФОРМАТИКА

Старший преподаватель департамента информационных технологий и автоматики
Шеклеин Алексей Александрович

Уральский федеральный университет

СЖАТИЕ

- Сжатие данных
- Кодирование по алгоритму Хаффмана
- Межсимвольная зависимость
- Избыточность в английском языке
- Кодирование Лемпеля – Зива
- Другие формы сжатия информации

Уральский федеральный университет

СЖАТИЕ ДАННЫХ

Сжатие данных (data compression) – процедура уменьшения их объёма с сохранением (полным или частичным) их целостности.

Применение: рациональное использование устройств хранения и передачи данных.

Синонимы: упаковка данных, компрессия, сжимающее кодирование, кодирование источника.

Обратная процедура называется восстановлением данных (распаковкой, декомпрессией, декодированием).

Уральский федеральный университет

Сжатие основано на устранении **избыточности**, содержащейся в исходных данных.

Примеры избыточности в тексте:

1. Часто встречающиеся значения данных. Например, буква e в английском алфавите.
2. Повторение в тексте фрагментов. Например:

THIS THESIS IS THE THESIS

(ЭТА ДИССЕРТАЦИЯ ИМЕННО ТА ДИССЕРТАЦИЯ)

Способы решения (устранения избыточности):

1. Замена часто встречающихся данных короткими кодовыми словами, а редких – длинными (энтропийное кодирование).
2. Замена повторяющейся последовательности ссылкой на уже закодированный фрагмент с указанием его длины.

Сжатие данных, **не обладающих свойством избыточности** (например, случайный сигнал или белый шум, зашифрованные сообщения), принципиально невозможно без потери целостности этих данных.

Существуют десятки методов сжатия, каждый из которых имеет свои преимущества и сферы применения.

Уральский федеральный университет

Все методы сжатия данных делятся на два основных класса:

- **Сжатие без потерь** (полное восстановление исходных данных) используется для передачи и хранения текстовых файлов.
- **Сжатие с потерями** (восстановление данных с искажениями) используется для сокращения объёма музыки, графики, видео, цифровых фотографий и т.п.

два способа сжатия текста (сжатие без потерь):

1) Кодирование Хаффмана (энтропийное кодирование)
2) Кодирование Лемпеля-Зива (LZ77)

Уральский федеральный университет

КОДИРОВАНИЕ ПО АЛГОРИТМУ ХАФФМАНА

Алгоритм Хаффмана – это один из первых методов, в основе которого лежит энтропийное кодирование (кодирование источника).

Хаффман решил задачу отыскания моментальных кодов наименьшей длины.

Мы знаем, что

$$\boldsymbol{H \leq L < H + 1}.$$

Методика Хаффмана создает код со средней длиной L, максимально близкий к энтропии источника H.

При этом **может существовать несколько кодов с той же самой минимальной средней длиной**.

Уральский федеральный университет

АЛГОРИТМ ХАФФМАНА

Алгоритм состоит из двух этапов:

1. Сначала символы источника **сокращаются** путем последовательного образования сложных символов до тех пор, пока не останется только два символа.
2. Затем кодирование затем осуществляется в обратном порядке путем **разделения** сложных символов и добавления знаков кода по два одновременно. Когда все сложные символы будут разделены, получаются кодовые слова для символов источника.

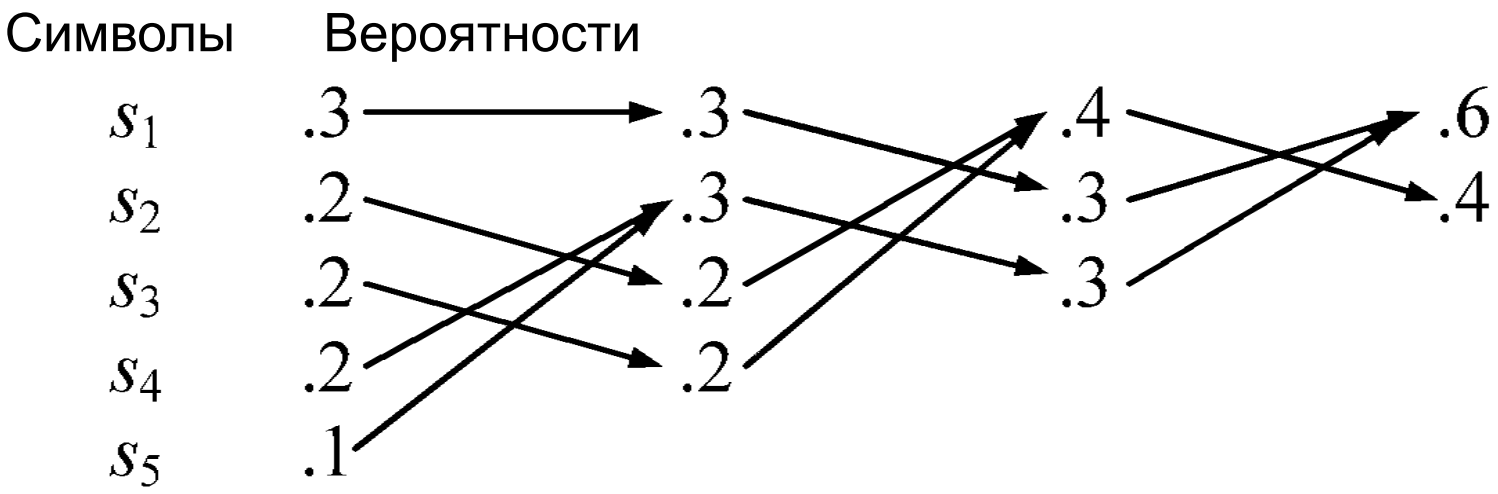
Уральский федеральный университет

ПРИМЕР 1

Рассмотрим источник из пяти символов с заданными вероятностями.

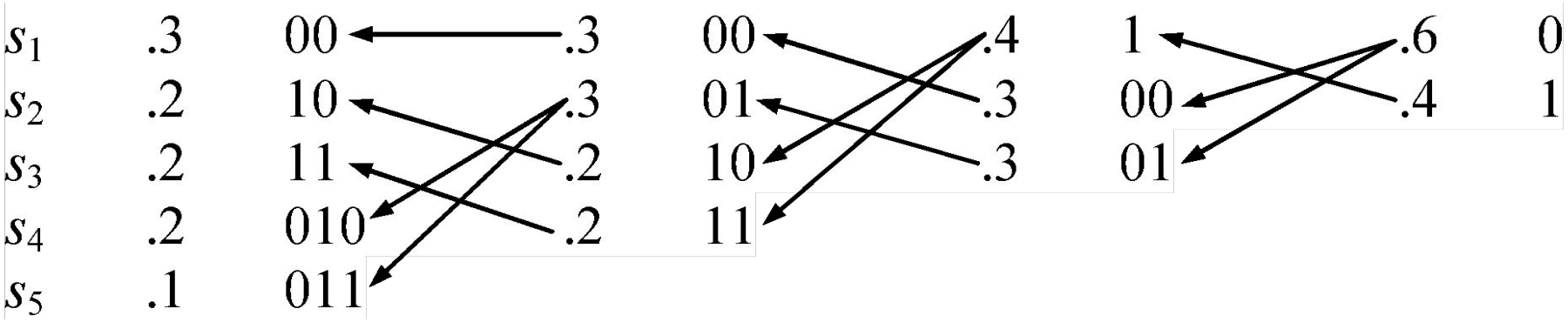
Символы	Вероятности
s_1	.3
s_2	.2
s_3	.2
s_4	.2
s_5	.1

1 этап – Сокращение. На каждом этапе два символа самой низкой вероятности объединяются, образуя один сложный символ. Новый список записывается в порядке убывания вероятностей. Это продолжается до тех пор, пока не останется два символа.



Уральский федеральный университет

2 этап – Разделение. Два символа в последнем списке кодируются с помощью 0 и 1. Затем они переносятся обратно в предыдущий список. Сложный символ при переносе должен быть разделен на те два символа, из которых он получен, и к существующему кодовому слову добавляется 0 и 1 соответственно для разделения символов.



Итак, мы получили код:

Символ источника	Вероятность	Код
s_1	0.3	00
s_2	0.2	10
s_3	0.2	11
s_4	0.2	010
s_5	0.1	011

Средняя длина кода:

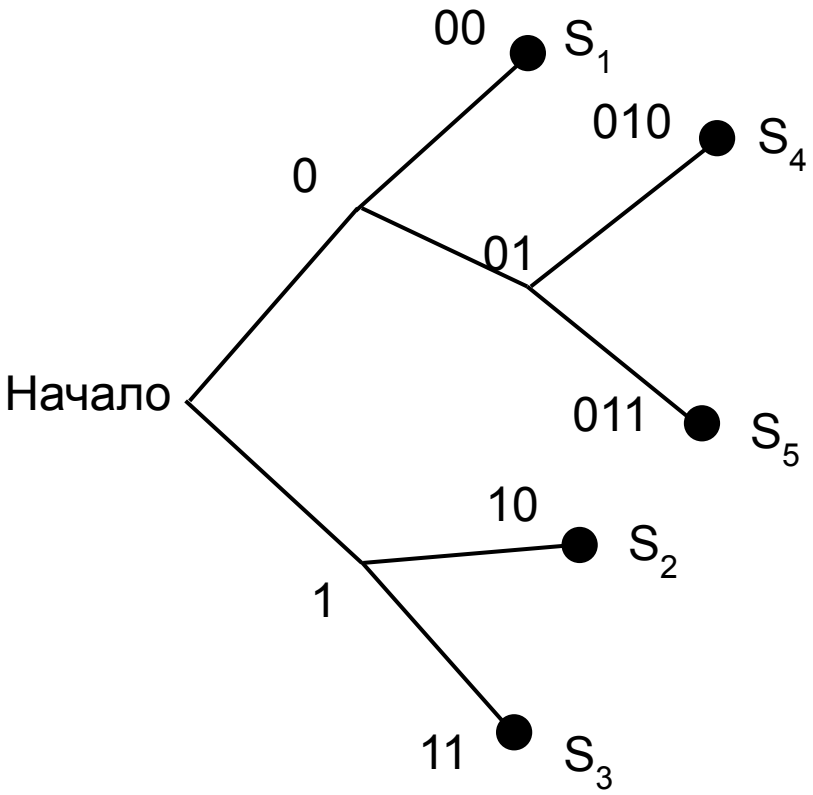
$$L = \sum_{i=1}^{m} p_i l_i = (2\times0,3) + (2\times0,2) + (2\times0,2) + (3\times0,2) + (3\times0,1) = 2,3$$

Энтропия источника:

$$H = -\sum_{i=1}^{m} p_i \log p_i = -[0,3\times\log 0,3 + (3 \times 0,2)\times\log 0,2 + 0,1\times\log 0,1] = 2,246$$

Уральский федеральный университет

Кодовое дерево Хаффмана для примера 1.



Уральский федеральный университет

ПРИМЕР 2

Неоднозначность кодов Хаффмана.

Попробуйте построить два различных кода Хаффмана для пятисимвольного источника.

Символ источника	Вероятность
s_1	0,4
s_2	0,2
s_3	0,2
s_4	0,1
s_5	0,1

Результат: два кода построены по методике Хаффмана и имеют одинаковую среднюю длину.

Символ источника	Вероятность	Код1	Код2
s_1	0,4	1	00
s_2	0,2	01	10
s_3	0,2	000	11
s_4	0,1	0010	010
s_5	0,1	0011	011

$$L_1 = (1\times0{,}4) + (2\times0{,}2) + (3\times0{,}2) + (4\times0{,}1) + (4\times0{,}1) = 2{,}2$$
$$L_2 = (2\times0{,}4) + (2\times0{,}2) + (2\times0{,}2) + (3\times0{,}1) + (3\times0{,}1) = 2{,}2$$

ПРИМЕР 3

Коды Хаффмана для S^n (степеней источника).

Рассмотрим источник S, состоящий из двух символов A и B с вероятностями $p_A = 3/4$, $p_B = 1/4$.

Энтропия источника S:

$$H = -\left[\frac{3}{4}\log\frac{3}{4} + \frac{1}{4}\log\frac{1}{4}\right] = -\left[\frac{3}{4}\log 3 + \frac{3}{4}\log\frac{1}{4} + \frac{1}{4}\log\frac{1}{4}\right] =$$

$$= \log 4 - (3/4)\log 3 = 2 - (3/4)\log 3 = 2 - 0{,}75 \times 1{,}5849 = 0{,}8118$$

Уральский федеральный университет

S			S^2			S^3		
Символ	Вероятность	Код	Символ	Вероятность	Код	Символ	Вероятность	Код
A	3/4	0	AA	9/16	0	AAA	27/64	1
B	1/4	1	AB	3/16	10	AAB	9/64	001
			BA	3/16	110	ABA	9/64	010
			BB	1/16	111	BAA	9/64	100
						ABB	3/64	00000
						BAB	3/64	00001
						BBA	3/64	00010
						BBB	1/64	00011
$L = 1$			$L/2 = 0{,}84375$			$L/3 = 0{,}8229167$		

По мере увеличения числа комбинаций средняя длина на один символ кода Хаффмана приближается к энтропии источника $H = 0{,}8118$.

Уральский федеральный университет

МЕЖСИМВОЛЬНАЯ ЗАВИСИМОСТЬ

Межсимвольная зависимость. Следующие друг за другом символы от источника не всегда являются независимыми. Наоборот, текущие вероятности символов часто зависят от того, какие символы встречаются перед ними.

Межсимвольная зависимость сокращает энтропию, и следовательно, могут быть построены коды с более короткими средними значениями длины слова.

Эта идея является основой многих современных методов сжатия.

Уральский федеральный университет

ВЕРОЯТНОСТИ В АНГЛИЙСКОМ ЯЗЫКЕ

Рассмотрим оценку фактической энтропии английского языка с учетом межсимвольной зависимости.

Будем считать, что в алфавите 26 символов и пробел (всего 27 символов).

Поскольку $2^5 = 32$, все 27 символов могут быть закодированы с помощью двоичного блок-кода $L = 5$.

Оценка энтропии английского языка нулевого порядка.

Предполагает, что все символы встречаются равновероятно, т.е. $p_i = \frac{1}{27}$. Поэтому

$$H_0 = \log 27 = 4{,}755 \text{ бита/букву}.$$

Вероятности встречаемости букв английского алфавита.

A	0,064	N	0,056
B	0,014	O	0,056
C	0,027	P	0,017
D	0,035	Q	0,004
E	0,100	R	0,049
F	0,020	S	0,056
G	0,014	T	0,071
H	0,042	U	0,031
I	0,063	V	0,010
J	0,003	W	0,018
K	0,006	X	0,003
L	0,035	Y	0,018
M	0,020	Z	0,002
	Пробел/пунктуация		0,166

Энтропия английского языка на основе вероятностей букв называется **оценкой первого порядка**. Для рассчитанных вероятностей:

$$H_1 = 4{,}194 \text{ бита/букву.}$$

Таким образом, код Хаффмана для алфавита будет иметь среднюю длину ближе к четырем, чем к пяти.

Кодирование по алгоритму Хаффмана действительно используется в качестве **метода компрессии английского текста**.

При этом документы требуют приблизительно на 20 % меньше памяти, чем в случае использования стандартного кода ASCII.

Уральский федеральный университет

ПРИМЕНЕНИЕ ЗАКОНОВ ЦИПФА (ЗИПФА)

Эксперимент. Не трудно понять искаженное слово scho_l (шко_а).

Шеннон предложил рассматривать частотности слов, а не частотности букв и рассчитал энтропию H_W слов английского языка.

Тогда энтропия на одну букву английского языка:

$$H = H_w/\bar{w}.$$

где $\bar{w}$ – это средняя длина английских слов.

Уральский федеральный университет

Для расчета Шеннон воспользовался законом Ципфа. (Джордж Ципф в 1949 г. эмпирически вывел соотношение частотностей слов).

Ципф заметил, что длинные слова встречаются в текстах любого языка реже, чем короткие.

Закон Ципфа. Если слова языка расположены в списке в порядке убывающей частотности (когда первое слово является самым частотным, а слово m занимает m-ю позицию в списке), в этом случае вероятности этих слов приблизительно удовлетворяют соотношению:

$$p_m = \frac{A}{m}$$

где A – это постоянная величина, зависящая от числа активных слов в языке.

Шеннон использовал:

$A = 0{,}01$ (постоянная величина для английского языка).

$M = 12366$ (количество слов в английском языке)

Тогда

$$\sum_{m=1}^{M} p_m = \sum_{m=1}^{12366} \frac{0{,}01}{m} = 1 \qquad H_W = \sum_{m=1}^{12366} \frac{0{,}01}{m} = 9{,}72 \quad \textit{бита / слово.}$$

Если средняя длина слова $\bar{w} = 4{,}5$ буквы/слово, то

$$H = 9{,}72/4{,}5 = 2{,}16 \text{ бита/букву.}$$

Это уменьшение более чем на 50% по сравнению с простой энтропией, основанной на 27

Уральский федеральный университет

ИЗБЫТОЧНОСТЬ АНГЛИЙСКОГО ЯЗЫКА

Эксперимент:

Отгадайте букву, которая идет дальше в следующем тексте:

She was so astonished and bewildered that sh_

(Она была так удивлена и обескуражена, что он_)

Предсказания будут верными в 75% случаев, а в остальных вероятность угаданной буквы равняется 1/26 (одна из 26 возможных букв алфавита).

Тогда энтропия неопределенности составит:

$$H = -0{,}75 \times \log 0{,}75 - 26 \times (0{,}25/26)\log(0{,}25/26)$$
$$= 1{,}99 \text{ бита}$$

Фактически даже **1,5** бита на букву!

Шеннон определил избыточность следующим образом:

$$R = 1 - H / \log M,$$

где M – размер алфавита, а H – истинное значение энтропии.

Отсюда для английского языка:

$$R \approx 1 - \frac{1{,}5}{\log 27} = 1 - 1{,}5 / 4{,}755 = 67\%$$

Выводы:

- Имеется огромная возможность для компрессии английского текста.
- Текстовый файл может **теоретически** быть спрессован до размера, составляющего одну треть от первоначального.
- **НО** для достижения результатов, близких к теоретическим, необходимо использовать более сложный процесс кодирования и декодирования, чем кодирование по алгоритму Хаффмана.

Уральский федеральный университет

КОДИРОВАНИЕ ЛЕМПЕЛЯ – ЗИВА

Новаторский и талантливый метод компрессии предложили Зив и Лемпель в 1977 г.

Метод LZ77 основывается на том, что последовательности букв в английском тексте содержат повторяющиеся фрагменты (образцы).

Сначала при передаче текста образцы накапливаются, а затем, продолжая передавать текст, отправитель уже смотрит на имеющийся словарь образцов и передает вместо самой последовательности **ссылку на имеющийся образец**.

В методе Лемпеля – Зива и отправитель, и получатель ведут запись того, что уже было отправлено.

Затем при подготовке к отправке дополнительного текста отправитель снова смотрит на ранее переданный текст, для того чтобы найти дублирование максимальной длины того, что требуется отправить дальше.

После этого вместо самой последовательности отправляется ссылка на прошлую дублирующую последовательность.

Например, если следующей частью текста является слово, которое передавалось раньше, отправитель просто отправляет ссылку на позицию этого слова в записи прошлых букв.

Ссылка на образец передается путем отправки тройки

$$(x, y, z),$$

где

x – это число позиций в тексте в обратном направлении до начала дублируемой последовательности,

y – длина дублируемой последовательности,

z – это следующая буква в предаваемом тексте после дублируемой последовательности.

Например, сообщение (5, 3, F)

ПРИМЕР. LZ77

Исходный текст:

THIS-THESIS-IS-THE-THESIS.

Передаваемые сообщения (код LZ77):

(0, 0, T)	T
(0, 0, H)	TH
(0, 0, I)	THI
(0, 0, S)	THIS
(0, 0, -)	THIS-
(5, 2, E)	THIS-THE
(5, 1, I)	THIS-THESI
(7, 2, I)	THIS-THESIS-I
(10, 5, -)	THIS-THESIS-IS-THE-
(14, 6, .)	THIS-THESIS-IS-THE-THESIS.

Уральский федеральный университет

Задания:

1) Закодируйте сообщение:

ROCCOBAROCCO.

Код LZ77:

(0,0,’R’) (0,0,’O’) (0,0,’C’) (1,1,’O’) (0,0,’B’) (0,0,’A’) (7,5,’.’)

2) Декодируйте сообщение

(0, 0, ‘a’) (0, 0, ‘b’) (0, 0, ‘r’) (3, 1, ‘c’) (2, 1, ‘d’) (7, 4,’’)

Сообщение:

abracadabra

Уральский федеральный университет

СОВРЕМЕННЫЕ АРХИВАТОРЫ

Существуют варианты базовой процедуры Лемпеля – Зива.

Проблема базовой процедуры Лемпеля – Зива: если текст большой, ссылки на образцы могут иметь большие значения x и y. Поэтому ссылки тоже подвергаются сжатию, например по методу Хаффмана.

Несколько популярных алгоритмов компрессии на персональных компьютерах, включая **ZIP**, **PKZIP**, **LHArc**, **PNG**, **gzip** и **ARJ** основаны на методе Лемпеля – Зива LZ77 вместе с компрессионным кодированием ссылочной информации.

Уральский федеральный университет

ДРУГИЕ ФОРМЫ СЖАТИЯ ИНФОРМАЦИИ

Кроме текста, сжатие также применяется к графике, изображениям, речи, видео…(эти данные не обладают избыточностью)

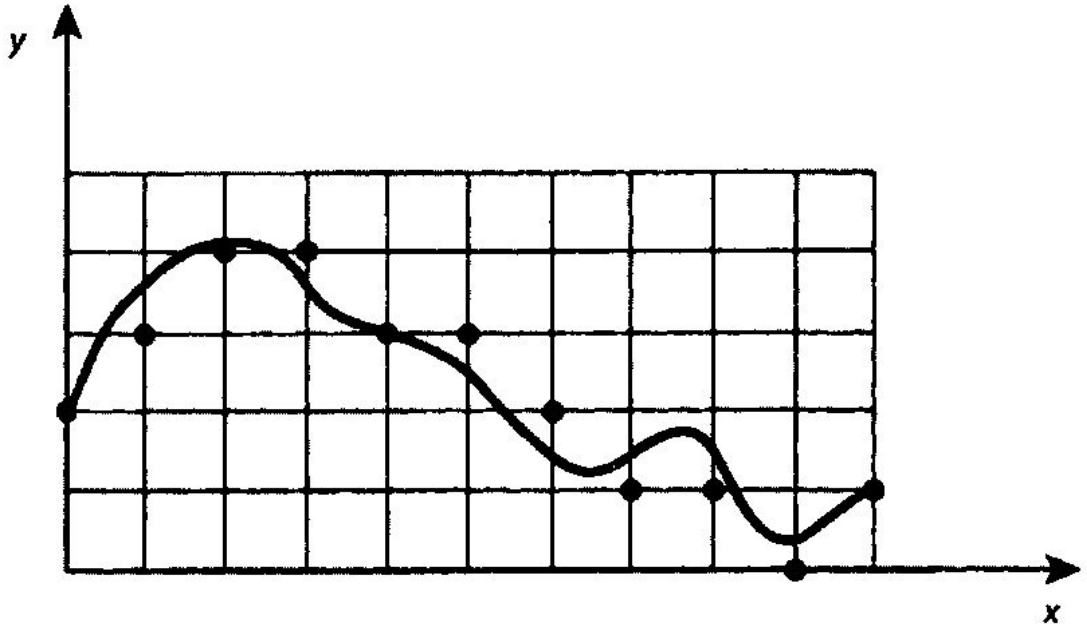
Здесь данные выражаются в виде численных значений. Это могут быть: интенсивность для речевых или музыкальных образов; интенсивность цвета, яркости (фотография) и т.п.

Общепринятой является обработка этих видов данных первоначально посредством **дискретизации** временного и пространственного измерения, а также уровней интенсивности.

Например, уровни речи могут записываться каждую миллисекунду с точностью в пределах восьми битов, а значения интенсивности изображения могут записываться посредством нескольких миллионов пикселей, при этом каждое обеспечивает 24 бита цветных данных.

Пример. Квантизация графика.

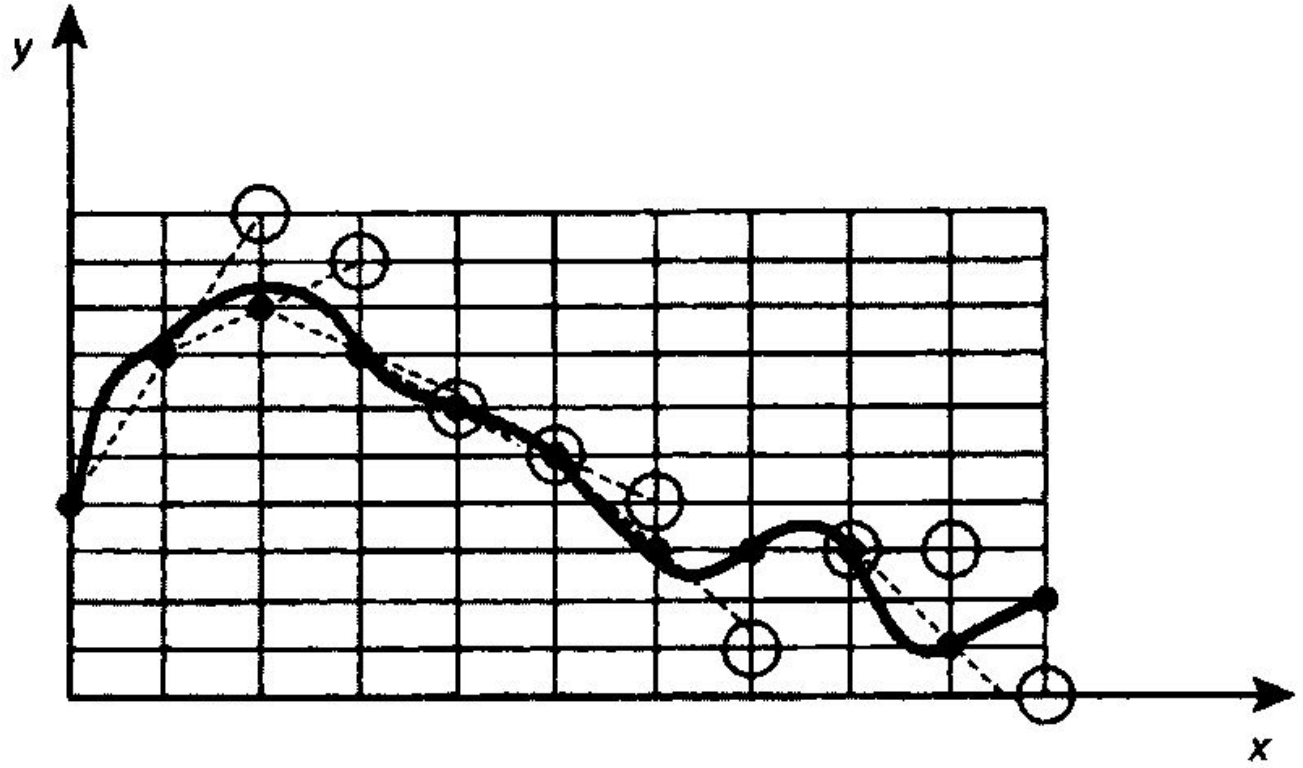
С регулярным интервалом на оси x значения y графика аппроксимируются до ближайшей точке сетки.



Чем мельче сетка, тем больше точек, и соответственно, больше бит.

МЕТОДЫ ПРЕДСКАЗАНИЯ

Квантизация с предсказанием. Предсказание y, представленное открытой окружностью, выполняется вычерчиванием пунктирной линии между двумя предыдущими точками и продолжением ее до следующего значения x. Затем требуется только передать погрешность предсказания.



Матрица яркости для изображения: в части *а* показаны значения яркости; в части *б* – элементы A, B, C, используемые для предсказания значения X.

14	12	10	18	20	24	30
12	12	14	16	18	26	32
12	14	12	16	20	24	30
18	10	10	12	16	40	50
12	12	14	35	40	55	60
14	45	48	50	56	114	80
50	52	46	55	110	116	92
55	56	50	58	112	108	94

(а) Матрица яркости

A	B
C	X

(б) Элементы предсказания

Подходящим предсказанием является следующее: $X = \mathrm{B} + \mathrm{C} - \mathrm{A}$.

Уральский федеральный университет

JPEG

Стандарты сжатия графической информации были установлены комитетом, который называется **Совместной группой экспертов по фотографии** (**Joint Photographic Experts Group**) (**JPEG**).

Данные стандарты используются в популярных пакетах для сжатия изображений JPEG.

Комитет разработал два основных метода: метод без потерь (рассмотренный выше метод предсказания) и метод с потерями (кратко рассматривается далее, имеет более значительное сжатие).

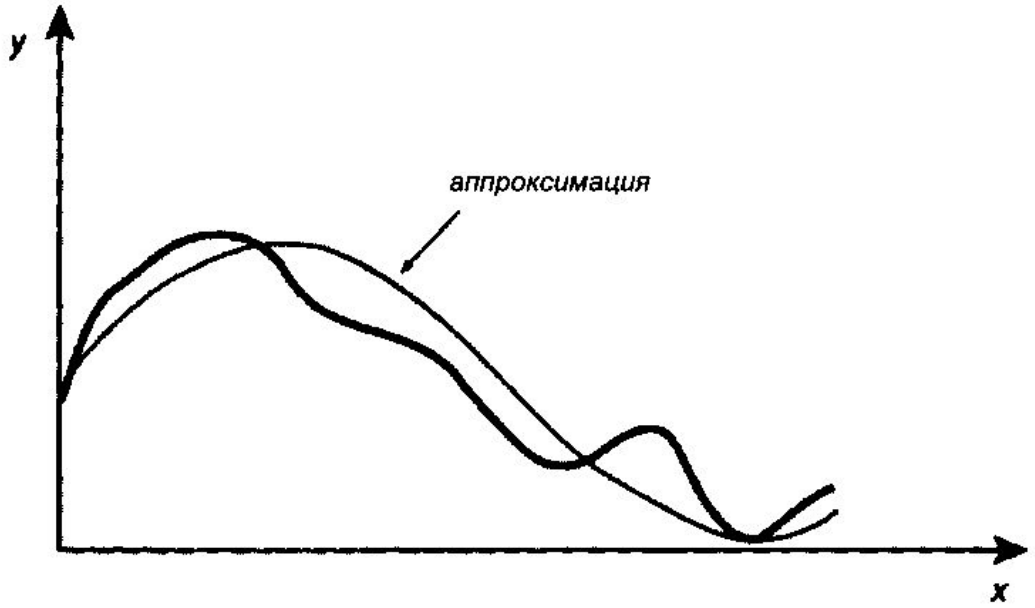
АППРОКСИМАЦИЯ

Аппроксимация посредством многочлена.

Первоначальная кривая аппроксимируется посредством кубического многочлена

$$a_3x^3 + a_2x^2 + a_1x + a_0 .$$

После этого требуется только передать коэффициенты этого многочлена, для того чтобы послать первоначальную кривую.



Популярная версия **JPEG** аппроксимирует матрицу яркости посредством серии **двухмерных функций косинуса** и записывает коэффициенты этой серии.

Метод разработан таким образом, что даже при коэффициенте сжатия 10:1 всего лишь незначительное ухудшение изображения будет заметно глазу человека.

MP3

MPEG (филиал комитета JPEG) – группа экспертов по киноизображениям разработала стандарты сжатия для киноизображений и высококачественных аудио сигналов.

Стандарт **MPEG1** способен компрессировать как видеосигналы, так и аудиосигналы.

Популярный стандарт компрессии музыки **MP3** фактически представляет собой аудиочасть стандарта MPEG1.

Уральский федеральный университет

Упражнения

1. (Шестисимвольный источник.) Источник имеет шесть символов с нижеуказанными вероятностями.

s_1	0,3	s_4	0,1
s_2	0,2	s_5	0,1
s_3	0,2	s_6	0,1

(a) Найти код Хаффмана для этого источника.

(b) Какова средняя длина кода Хаффмана?

(c) Какова энтропия источника?

2. (Пятисимвольный источник.) Найти три различных кода Хаффмана для источника.

s_1	0,4	s_4	0,1
s_2	0,2	s_5	0,1
s_3	0,2		

3. (Преимущество алгоритма Хаффмана.) Преимущество компрессии при кодировании по алгоритму Хаффмана S^2 вместо S является наиболее значительным, когда имеют место большие отличия в вероятностях символа источника. Для каждого из двух случаев, указанных ниже, найти энтропию и среднюю длину кода Хаффмана для S и S^2.

(a)	s_1	0,60
	s_2	0,40
(b)	s_1	0,90
	s_2	0,10

Уральский федеральный университет

4. (Вычитание Хаффмена.) Рассмотрите источник со связанными с ним символами и вероятностями

s_1	0,2	s_5	0,1
s_2	0,2	s_6	0,1
s_3	0,2	s_7	0,1
s_4	0,1		

Один код Хаффмена для данного источника имеет значения длины слова, равные 2, 3, 3, 3, 3, 3, 3. Не прибегая к использованию процедуры Хаффмена, найти мгновенный двоичный код минимальной средней длины для этого источника, имеющего эти значения длины слова.

5. (Пример LZ77.)

Декодируйте сообщение

(0, 0, ‘a’) (0, 0, ‘b’) (0, 0, ‘r’) (3, 1, ‘c’) (2, 1, ‘d’)

(7, 4,’’)

Уральский федеральный университет

Спасибо за внимание!