# Сжатие информации и энтропия

December 6, 2025

Сжатие данных - алгоритмическое преобразование данных. Путем применения алгоритмов и методов компрессии, мы можем существенно уменьшить размер данных, сохраняя при этом их суть и основные характеристики. Это применяется для более рационального использования устройств хранения и передачи данных.

## 1 Сжатие без потерь

История алгоритмов сжатия данных без потерь берёт своё начало в 1949 году, когда свет увидели разработки Клода Шеннона и Роберта Фано. Созданный ими алгоритм Шеннона - Фано заложил основы для будущих достижений в этой области: в его основе лежала идея использования вероятности появления символов для формирования кодов в блоке.

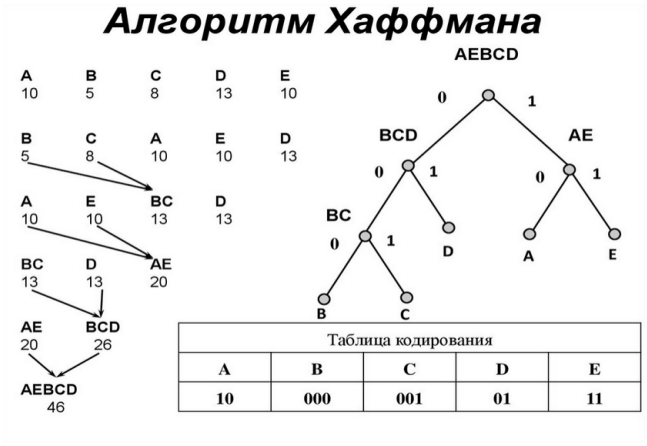
Спустя два года мир узнал о новом методе бинарного кодирования - коде Хаффмана. Его автором стал Дэвид Хаффман, который в то время был студентом и обучался в группе Роберта Фано в Массачусетском технологическом институте (MIT). Курсовая работа молодого учёного представила миру один из наиболее эффективных на тот момент способов кодирования.

С наступлением конца 1970-х годов компьютеры постепенно входили в обиход, и на свет появился первый программный продукт для сжатия данных. На первых порах в его основе лежал алгоритм Хаффмана, однако уже в 1977 году произошёл новый прорыв - был разработан алгоритм Лемпеля и Зива (LZ77). Эта разработка стала знаковой, поскольку впервые использовала словари для выявления повторяющихся последовательностей битов, что существенно повысило эффективность сжатия данных.

## 1.1 Алгоритм Хаффмана

Алгоритм Хаффмана - это способ создания оптимального префиксного кода, который помогает существенно сократить объём данных. Префиксное кодирование гарантирует, что ни один код не будет началом другого, благодаря чему декодирование становится однозначным, а данные — полностью сохраняемыми.

Принцип работы алгоритма начинается с тщательного анализа входных данных. Сначала вычисляется частота появления каждого символа в тексте. Затем на основе этих данных формируется так называемое дерево Хаффмана - особая двоичная структура. В этом дереве каждый узел хранит информацию о символе и его частоте, а конечные узлы, или листья, представляют отдельные символы.




Построение дерева - процесс последовательного объединения символов по их частотам. На каждом этапе выбираются два символа или узла с наименьшей частотой и объединяются в новый узел, частота которого равна сумме частот «родительских» узлов. Этот процесс продолжается до тех пор, пока не образуется единое дерево с одним корневым узлом.

Когда дерево готово, для каждого символа можно определить его префиксный код. Для этого необходимо проследить путь от корневого узла до нужного листа: движение к левому потомку отмечается как «0», а к правому - как «1». В итоге каждый символ получает уникальный код, который не совпадает с началом кода другого символа

Затем осуществляется кодирование: каждый символ в исходных данных заменяется на соответствующий префиксный код, что позволяет сократить объём информации. Для восстановления исходных данных используется то же дерево Хаффмана: кодовые слова преобразуются обратно в символы.

Алгоритм Хаффмана ценится за сравнительную простоту реализации и высокую эффективность при работе с данными, в которых символы распределены

неравномерно. Благодаря этим качествам он широко применяется в различных областях - от сжатия текстовых файлов до обработки аудиозаписей и изображений.

Однако у алгоритма есть и свои ограничения. Например, он не может обеспечить такое же сильное сжатие, как методы с потерями. Кроме того, эффективность Хаффмана сильно зависит от распределения частот символов: если все символы встречаются с одинаковой вероятностью, сжатие будет минимальным, а в некоторых случаях размер данных даже может увеличиться из-за необходимости хранить таблицу кодов.

## 1.2 Алгоритм LZ77

В последнюю декаду 1970-х годов вычислительные машины начали завоёвывать всё большую популярность, и в это же время появились первые программные решения для компрессии резервных копий данных. На начальных этапах в основе таких решений лежали коды Хаффмана, однако настоящий прорыв произошёл в 1977 году, когда Лемпель и Зив представили свой новаторский алгоритм (LZ77) - первую систему сжатия, задействующую словари.

Суть метода заключается в замене дублирующихся цепочек символов ссылками на их предыдущие появления в данных. В работе алгоритма задействованы две ключевые структуры: окно поиска и выходной буфер. В окне поиска хранятся уже обработанные данные, тогда как выходной буфер содержит ту часть информации, которая подвергается обработке в настоящий момент.

Когда алгоритм сталкивается с новой последовательностью символов, которой нет в окне поиска, он добавляет её в выходной буфер. В случае же, если такая последовательность уже встречалась, LZ77 фиксирует ссылку на её местоположение в окне поиска и указывает длину. Благодаря этому подходу удаётся существенно сократить объём данных, особенно если исходный материал изобилует повторяющимися фрагментами.

| Содержимое окна | Содержимое буфера | КОД |
|---|---|---|
| $kabababababz$ | $k$ | <0,0,$k$> |
| $kabababababz$ | $a$ | <0,0,$a$> |
| $kabababababz$ | $b$ | <0,0,$b$> |
| $k$ ab $ababababz$ | $aba$ | <2,2,$a$> |
| $kaba$ babab $abz$ | $bababz$ | <2,5,$z$> |

Этот алгоритм лёг в основу множества современных стандартов сжатия данных, включая DEFLATE, который применяется в таких форматах, как PNG и ZIP. Среди сильных сторон LZ77 - относительная простота внедрения и высокая эффективность сжатия для текстов, содержащих многочисленные повторы.

## 2 Сжатие с потерями

Сжатие данных с потерями - это метод уменьшения размера файла путем исключения части информации, которая считается менее значимой.

Алгоритмы сжатия данных с потерями находят применение в ситуациях, когда допустимо жертвовать небольшим объёмом информации ради существенного уменьшения размера файлов. Подобные методы особенно востребованы в сферах, где критически важна оптимизация хранения и передачи данных, например, при работе с мультимедийным контентом, аудио- и видеофайлами.

Процедура сжатия с потерями состоит из нескольких важных стадий:

- На первом этапе проводится детальный анализ исходных данных, в ходе которого выявляются элементы, не являющиеся критически важными для восприятия конечного контента.
- Затем происходит устранение избыточной информации - удаляются данные, мало влияющие на общее качество материала, что позволяет существенно сократить объём файла.
- На завершающей стадии оставшаяся информация подвергается кодированию с применением специализированных алгоритмов, которые позволяют достичь максимальной компактности данных.

Среди наиболее распространённых алгоритмов сжатия с потерями можно выделить:

- JPEG - используется для сжатия изображений. Алгоритм преобразует изображение в особое цветовое пространство, разделяя информацию на компонент яркости и компоненты цвета, и преимущественно сжимает цветовые данные. Такой подход хорошо работает с фотографиями, поскольку человеческий глаз не слишком чувствителен к потерям цветовых деталей.
- MP3 - применяется для сжатия аудио. В основе алгоритма лежит психоакустическая модель, которая помогает идентифицировать и удалять звуковые частоты, не воспринимаемые человеческим ухом либо замаскированные более громкими звуками.
- MP4 - используется для сжатия видео. Алгоритм комбинирует методы покадрового и межкадрового сжатия: кадр разбивается на блоки, сохраняются только те данные, которые изменились между кадрами, также может снижаться битрейт, разрешение и частота кадров.

Стоит отметить, что попытка дополнительно сжать файлы, уже обработанные алгоритмами типа JPEG, MP3 или MP4, с помощью архиваторов не приведёт к значительному уменьшению их размера. Причина в том, что такие файлы уже подверглись интенсивному сжатию и практически не содержат избыточной информации на момент сохранения в своём формате.

Источники:

https://habr.com/ru/companies/otus/articles/745628/
https://habr.com/ru/companies/ruvds/articles/712652/
https://habr.com/ru/articles/141827/
https://ya.zerocoder.ru/pgt-kak-ustroeno-szhatie-s-poteryami/
https://docstech.ru/huffman-algorithm/

Более глубокие исследования:

"A Method for the Construction of Minimum-Redundancy Codes" (1952) - оригинальная статья Д.Хаффмана

"A Universal Algorithm for Sequential Data Compression" (1977) - оригинальная статья А.Лемпеля и Я.Зива

"Методы сжатия данных. Устройство архиваторов, сжатие изображений и видео." (2002) - Д. Ватолин, А. Ратушняк, М. Смирнов, В. Юкин.