Санкт-Петербургский Государственный Университет
Аэрокосмического Приборостроения

Курсовая Работа

Алгоритм Фано

Преподаватель Иванов Д.О.
Студент Сирота А.В.

Санкт-Петербург 2016

Содержание

1 Постановка задачи

Целью этой курсовой работы является разработка программы, которая будет иметь следующий функционал:

1. Кодирование файла
    - Побайтовый анализ кодируемого файла
    - Построение кодов кодирования по алгоритму Фано
    - Создание закодированного файла
2. Декодирование файла
    - Анализ закодированного файла
    - Создание раскодированного файла, который полностью идентичен файлу до кодировки

2 Алгоритм

Алгоритм Шеннона — Фано — один из первых алгоритмов сжатия, который впервые сформулировали американские учёные Шеннон и Роберт Фано. Он был независимо друг от друга разработан Шенноном (публикация «Математическая теория связи», 1948 год) и, позже, Фано (опубликовано как технический отчёт).

2.1 Основные сведения

Алгоритм Шеннона — Фано относится к вероятностным методам сжатия. *Вероятностный метод сжатия* — это метод, основанный на частоте встречаемости символа в тексте.

Подобно алгоритму Хаффмана, алгоритм Шеннона — Фано использует избыточность сообщения, заключённую в неоднородном распределении частот символов его первичного алфавита, то есть заменяет коды более частых символов короткими двоичными последовательностями, а коды более редких символов — более длинными двоичными последовательностями.

Коды, построенные по этому алгоритму, являются префиксными. Это необходимое условие однозначного декодирования. Префиксным называют код, удовлетворяющий условию Фано: если в код входит слово a, то для любой непустой строки b слова ab в коде не существует.

Например, код, состоящий из слов 0, 10 и 11, является префиксным, и сообщение

$$01001101110$$

можно разбить на слова единственным образом:

$$0 \quad 10 \quad 0 \quad 11 \quad 0 \quad 11 \quad 10$$

Код, состоящий из слов 0, 10, 11 и 100, префиксным не является, и то же сообщение можно трактовать несколькими способами:

$$0 \quad 10 \quad 0 \quad 11 \quad 0 \quad 11 \quad 10$$

$$0 \quad 100 \quad 11 \quad 0 \quad 11 \quad 10$$

Префиксные коды наглядно могут быть представлены с помощью кодовых деревьев. Если ни один узел кодового дерева не является вершиной данного кода, то он обладает свойствами префикса. Узлы дерева, которые не соединяются с другими, называются конечными. Комбинации, которые им соответствуют, являются кодовыми комбинациями префиксного кода.

2.2 Основные этапы

Так выглядит алгоритм Фано в общем виде:

1. Символы первичного алфавита m1 выписывают по убыванию вероятностей.
2. Символы полученного алфавита делят на две части, суммарные вероятности символов которых максимально близки друг другу.
3. В префиксном коде для первой части алфавита присваивается двоичная цифра «0», второй части — «1».
4. Полученные части рекурсивно делятся и их частям назначаются соответствующие двоичные цифры в префиксном коде.

Пример построения кодов по алгоритму Фано показан на рис. 1. Пример построения кодов.

Элемент	p(el)	Шаг 1	Шаг 2	Шаг 3	Шаг 4	Итого
a	0.36	0	0			00
b	0.18		1			01
c	0.18	1	0			10
d	0.12		1	0		110
e	0.09			1	0	1110
f	0.07				1	1111

Рис. 1: Пример построения кодов

2.3 Алгоритм вычисления кодов

Код Шеннона — Фано строится с помощью дерева. Построение этого дерева начинается от корня. Всё множество кодируемых элементов соответствует корню дерева (вершине первого уровня). Оно разбивается на два подмножества с примерно одинаковыми суммарными вероятностями.

Эти подмножества соответствуют двум вершинам второго уровня, которые соединяются с корнем. Далее каждое из этих подмножеств разбивается на два подмножества с примерно одинаковыми суммарными вероятностями. Им соответствуют вершины третьего уровня.

Если подмножество содержит единственный элемент, то ему соответствует концевая вершина кодового дерева; такое подмножество разбиению не подлежит.

Подобным образом поступаем до тех пор, пока не получим все концевые вершины. Ветви кодового дерева размечаем символами 1 и 0.

На рис. 2 представлен пример построения кодового дерева. Каждой букве в соответствие поставлена встречаемость в кодируемом тексте.

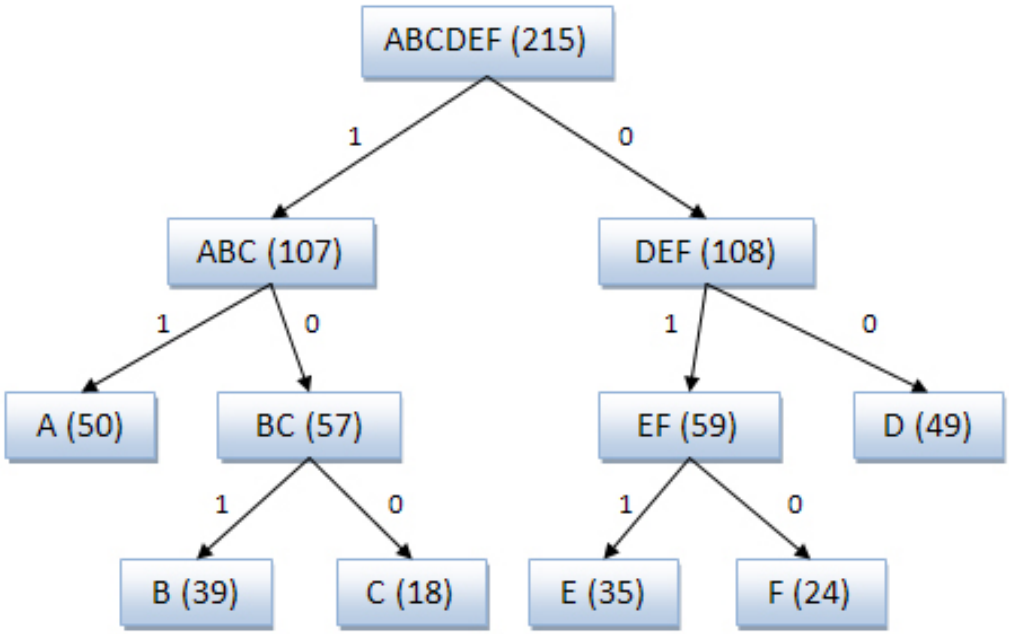

Рис. 2: Построение кодового дерева

Ниже представлен псевдокод рекурсивной функции, которая осуществляет построение бинарного дерева и вычисление кодов.

```
createCodeTree(begin, end, elements){
  //Если множество содержит один элемент, ничего не делать
  if(end == begin){
    return;
  }

  //Найти точку деления группы на подгруппы
  int point = findDividePoint(begin, end);

  //Добавить элементам первой подгруппы 0 в код
  for(int i = begin; i < point; i++){
    elements[i].code += "0";
  }

  //Добавить элементам второй подгруппы 1 в код
  for(int i = point; i < end; i++){
    elements[i].code += "1";
  }

  //Если в подгруппах > 1 элемента, делить дальше
  if(begin + 1 < point){
    createBranch(begin, point);
  }
  if(point + 1 < end){
    createBranch(point, end);
  }
}
```

При построении кода Шеннона — Фано разбиение множества элементов может быть произведено, вообще говоря, несколькими способами. Выбор разбиения на уровне n может ухудшить варианты разбиения на следующем уровне (n + 1) и привести к неоптимальности кода в целом.

Другими словами, оптимальное поведение на каждом шаге пути ещё не гарантирует оптимальности всей совокупности действий. Поэтому код Шеннона — Фано не является оптимальным в общем смысле, хотя и дает оптимальные результаты при некоторых распределениях вероятностей.

Для одного и того же распределения вероятностей можно построить, вообще говоря, несколько кодов Шеннона — Фано, и все они могут дать различные результаты. Если построить все возможные коды Шеннона — Фано для данного распределения вероятностей, то среди них будут находиться и все коды Хаффмана, то есть оптимальные коды.

Для примера, рассмотрим последовательность в табл. 1. Метод Хаффмана сожмёт её до 77 бит, а вот Шеннона — Фано до 79 бит.

Символ	Встречаемость	Код Хаффмана	Код Шеннона — Фано
a	14	0	00
b	7	111	01
c	5	101	10
d	5	110	110
e	4	100	111

Таблица 1: Сравнение кодов Хаффмана и Шеннона — Фано

Режим	Описание	Входной файл
-e	Режим кодирования. В выходной файл записывается шапка с таблицей кодировки и закодированный файл.	Любой
-d	Режим декодирования. Выходной файл является точной копией файла, который был закодирован.	*.fano

Таблица 2: Режимы работы программы

3 Руководство пользователя

Программа fano осуществляет сжатие текстовых файлов по алгоритму Шеннона – Фано.

Репозиторий на GitHub: https://github.com/nukefluke/fano-algorithm

Программа запускается из консоли, команда запуска выглядит так:

```
$ fano <mode> <input> [-o <output>]
```

Параметр mode отвечает за режим работы программы. Список режимов приведён в табл. 2.

Параметром input указывается путь ко входному файлу.

Параметр output устанавливает имя выходного файла. Если параметр не указан, имя файла будет сгенерировано автоматически.

3.1 Примеры команд запуска

Закодировать файл input.txt. Выходной файл будет иметь название input.txt.fano: $ fano -e input.txt Декодировать файл input.fano и сохранить как decoded.dat: $ fano –d input.fano –o decoded.dat

3.2 Производительность

3.3 Эффективность

Результаты тестов программы представлены в