УДК 621.391
С. В. Гринюк
Луцький національний технічний університет

Стиснення даних без втрат інформації на основі BWT – перетворення

В роботі розглядається використання перетворень в стисненні даних – перетворення потоку вхідних подій до вигляду, що дозволяє використовувати простіші і ефективніші моделі. До таких перетворень відносять і перетворення BWT (Burrows-Wheeler Transform), яке розглядається в даній роботі.
Ключові слова: перетворення BWT, пертворення MTF, стиснення даних.

В даний час спостерігається швидке збільшення кількості інформації, що зберігається передається та обробляється. Прогрес в галузі технічних засобів передачі та збереженні інформації
не вситагає за потребами людства в інформації. Запровадження в дію нових високопродуктивних комунікаційних систем коштує досить дорого.

Тому важливо з максимальною ефективністю використовувати наявні системи збереження і передачі інформації. Для цього потрібно подавати наявну інформацію меншою кількістю даних за рахунок використання ущільнення (стиснення) даних без втрат інформації – кодування інформації з мінімальною інформаційною надмірністю. Це дозволяє зберігати більше інформації на тому ж носії, передавати більше інформації за одиницю часу по каналу зв'язку тієї ж пропускної здатності.

Таким чином, очевидна економічна вигода від оптимізації подання інформації та актуальність розробки ефективних методів ущільнення даних.

Теоретичною основою ущільнення даних служать теорія інформації і теорія кодування. Родоначальником теорії інформації є К.Шеннон (C.E.Shannon). Важливі теоретичні результати в цій галузі належать А.М.Колмогорову, О.Я.Хінчіну, І.М.Гельфанду, А.Файнстейну (A.Fienstein), Дж.Вольфовіцу (J.Wolfowitz), Л.Бріллюену (L.Brillouin), П.Елайєсу (P.Elias), Р.Фано (R.M.Fano), Б.Мак-Міллану (B.McMillan).

Перший практичний метод ущільнення даних був запропонований незалежно один від одного К.Шенноном, Р.Фано і (у трохи зміненому вигляді) Д.Хаффманом (D.A.Huffman). Найбільший внесок до розробки практичних методів ущільнення даних внесли Дж.Ріссанен (J.J.Rissanen), Дж.Ленгдон (G.G.Langdon), Дж.Зів (JZiv), А.Лемпел (A.Lempel), Т.Белл (T.C.BeII), І.Віттен (I.H.Witten), Дж.Кпірі (J.G.CIeary).

Однак у теорії і практиці ущільнення даних без втрат лишилося чимало невирішених проблем Не надана загальна класифікація існуючих алгоритмів ущільнення з врахуванням особливостей методів побудови моделей джерел інформації, що лежать у їх основі Не запропоновано досить зручної універсальної оцінки стискаючої здатності методів. Недостатньо досліджені можливості побудови моделей і алгоритмів, що добре пристосовуються до різких змін характеру вхідних даних Остання проблема особливо актуальна у світі тенденції, що з'явилася в сучасних комп'ютерних системах-комбінування в одному файлі даних різних типів (текст, аудюдані, графічна інформація та ін.), що служать для вирішення однієї задачі.

Ущільнення зображень з втратами включає використання методів ущільнення без втрат на
останньому етапі, який і виконує власне ущільнення і від якого в значній мірі залежить загальний коефіцієнт ущільнення зображення. Однак, з появою методу арифметичного кодування проблема генерації коду була фактично вирішена. З тих пір з метою підвищення коефіцієнту ущільнення основна увага стала приділятися питанням, пов'язаним з моделюванням. Нові підходи опираються на парадигму ущільнення за допомогою універсального моделювання і кодування, запропоновану Ріссаненом і Ленгдоном.

В світлі концепції універсального моделювання і кодування заслуговують на увагу методи
ущільнення без втрат на основі перетворень. Мета використання перетворень в ущільненні даних – перетворення потоку вхідних подій до вигляду, що дозволяє використовувати простіші і ефективніші моделі. Фактично, вони перетворюють одні види надмірності в інші, простіше модельовані. Тобто, перетворення дозволяє представляти оброблювану інформацію в

особливій формі, ідеально відповідній для подальшого ефективного кодування. Незвичність підходу полягає в наявності фактично двох етапів моделювання: перший етап – це робота перетворення, направлена на отримання «зручного» інформаційного представлення, а другий – побудова допоміжної моделі, на основі якої буде закодовано дане представлення.

До таких перетворень відносять перетворення MTF (Move To Front) та перетворення BWT
(Burrows-Wheeler Transform).

Алгоритм стиснення даних на основі перетворення Барроуза-Уїлера (Burrows-Wheeler Transform, далі BWT, сортування блоку даних) вперше був описаний порівняно недавно - в 1994 році. Він був опублікований 10 травня в статті "A Block-sorting Lossless Data Compression Algorithm" [1]. Хоча стверджується, що один з його авторів, Майкл Уїлер, придумав його набагато раніше, в 1983 році, але тоді не надав йому належного значення.

Зараз цей метод стрімко набуває популярності серед дослідників в області стиснення даних.
Цей метод привабливий своєю простотою і елегантністю.

Алгоритм являв собою сукупність трьох методів:

- Метод сортування блоку даних (власне який і називається перетворенням Барроуза-Уїлера),
- MoveToFront-перетворення (відоме також, як метод переміщення стопки книг),
- Простий статистичний кодер для стиснення перетворених на перших двох етапах даних.

Однак, якщо MTF давно використовується при ущільненні як в якості перетворення так і в якості самостійного методу ущільнення, то по-перше перетворення BWT може використовуватись тільки в якості перетворення, а по-друге за рахунок використання перетворення BWT сумісно з MTF можна досягнути значних коефіцієнтів ущільнення, особливо високочастотних компонент зображення. Перетворення BWT застосовується для перетворення ланцюжкової надмірності в надмірність повторення подій. Спочатку вхідний потік подій циклічно зсувається на одну позицію і
записується під початковим вхідним потоком стільки раз, скільки подій у вхідному потоці. Отримана квадратна матриця сортується по рядках зліва направо. Доведено, що для відновлення початковогопотоку подій достатньо останнього стовпця матриці (так званого префіксного стовпця) і номера рядка початкового потоку подій після сортування. Префіксний стовпець володіє великою надмірністю повторення подій і локальною надмірністю розподілу імовірності.

Однак, виконання зворотного перетворення BWT вимагає значних затрат пам'яті, особливо при
великих об'ємах вхідного блоку даних. Для швидкого зворотного перетворення додатково до власнеданих потрібний вектор зворотного перетворення, що є масивом чисел, розмір якого рівний числу символів в блоці. В роботі запропоновано алгоритм реалізації прямого і зворотного BWT перетворення, який ґрунтується на зберіганні в пам'яті лише чотирьох стовпців початкової матриці.

Постановка проблеми**.** Розробка удосконалених методів стиснення даних без втрати інформації, які відрізняються від існуючих підвищеною стискаючою здатністю на більшості класів даних, які підлягають стисненню без втрат, та поліпшеною адаптивністю до даних.

Мета і задачі дослідження. Основною задачею є розробка удосконалених методів ущільнення даних без втрат, що мають підвищену, у порівнянні з існуючими методами, стискаючу здатність на основних класах даних, яки підлягають ущільненню без втрат інформації.
Метою роботи є підвищення стискаючої здатності методів ущільнення даних без втрат та поліпшення їх адаптивності до різких змін статистичних властивостей даних, що ущільнюються.

Для досягнення поставленої мети у роботі сформульовані та вирішені наступні взаємопов'язані задачі дослідження:

- Визначити найбільш перспективні з наявних методів ущільнення даних без втрат, на підставі аналізу варіантів їхньої реалізації і наявних недоліків запропонувати удосконалені методи та алгоритми ущільнення;
- Запропонувати зручний алгоритм для ущільнення даних без втрат на основі сортування блоку даних.

Розробка алгоритму виконання прямого і зворотного BWT – перетворення

Перетворення BWT не стискає дані, але перетворює блок даних у формат, виключно підходящий для компресії. Розглянемо його роботу на спрощеному прикладі. Нехай є словник V з N символів. Циклічно переставляючи символи в словнику вліво, можна отримати N різних рядків довжиною N кожна. За допомогою прикладу перетворення рядка символів «абракадабра». Далі потрібно з рядка даних створити матрицю всіх можливих його циклічни перестановок. Першим рядком матриці буде початкова послідовність, другим рядком - вона ж, зсунута на один символ вліво, і т.д. Таким чином, отримаємо матрицю, зображену нижче:

0 абракадабра
1 бракадабраа
2 ракадабрааб
3 акадабраабр
4 кадабраабра
5 адабраабрак
6 дабраабрака
7 абраабракад
8 браабракада
9 раабракадаб
10 аабракадабр

Оскільки, дані поступають з файлу побайтно і якщо відомий розмір блока BWT-перетворення, то
немає сенсу очікувати прийому всього блоку над яким виконується перетворення. Можна сформувати матрицю циклічних перестановок на етапі читання файлу. Прийнятий байт спочатку записується в порядку прийому в "0" рядок матриці, а потім записується в інші рядки матриці в позиції, які визначаються за наступною формолою:

$$P = (rbwt - ja + ia) \bmod rbwt$$

де rbwt – розмір блоку BWT – перетворення, ia – номер поточного стовпця, ja – номер рядка в який записується черговий байт.

Відсортуємо всі рядки даної матриці у відповідності з лексикографічним порядком символів. Вважатимемо, що один рядок повинен знаходитися в матриці вище за інший в тому випадку, якщо в найлівішій з позицій, починаючи з якої рядки відрізняються, в цьому рядку знаходиться символ лексикографічно менший, ніж у іншого рядка. Іншими словами, слід відсортувати символи спочатку по першому символу, потім рядки, у яких перші символи рівні, - по другому і т.д.

0 аабракадабр
1 абраабракад
2 абракадабра – початковий рядок
3 адабраабрак
4 акадабраабр
5 браабракада
6 бракадабраа
7 дабраабрака
8 кадабраабра
9 раабракадаб
10 ракадабрааб

Тепер залишився останній крок - виписати символи останнього стовпця і запам'ятати номер початкового рядка серед відсортованих. Отже, «рдакрааааббб», 2 – це результат, отриманий в результаті перетворення Барроуза - Уілера.

Слід зазначити основну властивість перетворення. Оскільки здійснювалися саме циклічні перестановки, символи останнього стовпця передують початковим символам рядків, які

найбільшою мірою брали участь у сортуванні. Таким чином, якщо у вихідному файлі є два схожі рядки, то символи, що передують їм обом, будуть знаходитися поблизу один від одного в блоці, отриманому в результаті
перетворення. І чим більше ці рядки схожі, тим більша ймовірність того, що ці символи будуть знаходитися поруч.

Тут і далі рядки, наступні у вхідному блоці за символами з блоку вихідного, будуть називатися контекстами.

Розглянемо процес відновлення початкової матриці. Хай нам відомий тільки результат
перетворення, тобто - "рдакрааааббб", 2. Відсортуємо всі символи останнього стовпця у відповідності з лексикографічним порядком.

0 а
1 а
2 а
3 а
4 а
5 б
6 б
7 д
8 к
9 р
10 р

Очевидно, що в результаті такого сортування ми отримали перший стовпець початкової матриці. Оскільки останній стовпець відомий, додамо його в отриману матрицю:

0 а.....р
1 а.....д
2 а.....а
3 а.....к
4 а.....р
5 б.....а
6 б.....а
7 д.....а
8 к.....а
9 р.....б
10 р.....б

Тепер самий час пригадати, що рядки матриці були отримані в результаті циклічного зсуву початкового рядка. Тобто, символи останнього і першого стовпців утворюють один з одним пари. І нам ніщо не може перешкодити відсортувати ці пари, оскільки обов'язково існують такі рядки в матриці, які починаються з цих пар. І ще допишемо в матрицю і відомий нам останній стовпець.

Таким чином, два стовпці нам вже відомі. Легко помітити, що відсортовані пари разом з
символами останнього стовпця складають трійки. Аналогічно відновлюється вся матриця. А на підставі записаного наперед номера початкового рядка в матриці - і сам початковий рядок:

0 аа.....р ааб.....р аабр.....р аабракада.р аабракадабр
1 аб.....д абр.....д абра.....д абраабрак.д абраабракад
2 аб.....а абр.....а абра.....а абракадаб.а абракадабра
3 ад.....к ада.....к адаб.....к адабраабр.к адабраабрак
4 ак.....р ака.....р акад.....р акадабраа.р акадабраабр
5 бр.....а бра.....а браа.....а ... браабрака.а браабракада
6 бр.....а бра.....а брак.....а бракадабр.а бракадабраа
7 да.....а даб.....а дабр.....а дабраабра.а дабраабрака
8 ка.....а кад.....а када.....а кадабрааб.а кадабраабра

9 ра.....б раа.....б рааб.....б раабракад.б раабракадаб
10 ра.....б рак.....б рака.....б ракадабра.б ракадабрааб

Після того, як вдалося наочно показати принципову можливість зворотного перетворення, прийшов час визнати, що насправді немає необхідності відтворювати посимвольно всі рядки матриці по одному символу. Зверніть увагу, що при кожному прояві досі невідомого стовпця виконувалися одні й ті ж дії. А саме, з рядка, що починається з деякого символу останнього стовпця виходив рядок, в якій цей символ знаходиться на першій позиції. З рядка 0 виходить рядок 9, з 1 - 7 і т.п.:

0 а.....р 9
1 а.....д 7
2 а.....а 0
3 а.....к 8
4 а.....р 10
5 б.....а 1
6 б.....а 2
7 д.....а 3
8 к.....а 4
9 р.....б 5
10 р.....б 6

Для отримання вектора зворотного перетворення, визначимо порядок отримання символів першого стовпця із символів останнього:

2 а.....а 0
5 б.....а 1
6 б.....а 2
7 д.....а 3
8 к.....а 4
9 р.....б 5
10 р.....б 6
1 а.....д 7
3 а.....к 8
0 а.....р 9
4 а.....р 10

Останній стовпець чисел і є вектором зворотного перетворення. Тепер отримати початковий рядок зовсім просто. Насамперед візьмемо елемент вектора зворотного перетворення, відповідний номеру початкового рядка в матриці циклічних перестановок, T[2]=6. Інакше кажучи, як перший символ в початковому рядку слід узяти шостий символ з нульового стовпця "рдакрааааббб". Це символ "а". Далі T[6]=10. Це десятий символ з нульового стовпця "рдакрааааббб" - "б". T[10]= 4 - "р", T[4]=8 - "а", T[8]=3 - "к", T[3]= 7 - "а", T[7]= 1 - "д", T[1]= 5 - "а", T[5]= 9 - "б", T[9]= 0 - "р", T[0]= 2 - "а". В результаті отримаємо слово "абракадабра", що і потрібно.

1. Балашов К.Ю. Сжатие информации: анализ методов и подходов. – Минск, 2000. – 42 с (Препринт / Ин-т техн. Кибернетики НАН Беларуси; № 6).
2. Ватолин Д., Ратушняк А., Смирнов М., Юкин В. Методы сжатия данных. Устройство архиваторов, сжатие изображений и видео. - М.: ДИАЛОГ-МИФИ, 2003. - 384 с.
3. Семенюк В. В. Экономное кодирование дискретной информации. – СПб.: СПбГИТМО (ТУ),
2001. – 115 с.