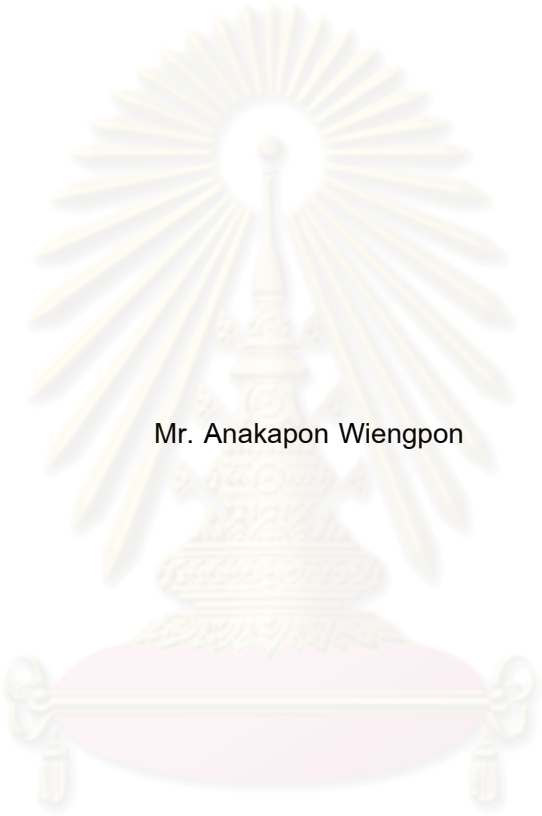
การปรับปรุงการเข้ารหัสเลขคณิตด้วยการจัดกลุ่มความน่าจะเป็นสำหรับเอกสารภาษาอังกฤษ

นายอนรรฆพล เวียงพล

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชาวิศวกรรมคอมพิวเตอร์ ภาควิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2553

AN IMPROVEMENT OF ARITHMETIC CODING USING PROBABILITY CLUSTERING FOR ENGLISH DOCUMENT

Mr. Anakapon Wiengpon



A Thesis Submitted in Partial Fulfillment of the Requirements
for the Degree of Master of Computer Engineering Program in Computer Engineering
Department of Computer Engineering
Faculty of Engineering
Chulalongkorn University
Academic Year 2010

หัวข้อวิทยานิพนธ์	การปรับปรุงการเข้ารหัสเลขคณิตด้วยการจัดกลุ่มความน่าจะเป็นสำหรับเอกสารภาษาอังกฤษ
โดย	นายอนรรฆพล เวียงพล
สาขาวิชา	วิศวกรรมคอมพิวเตอร์
อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก	ผู้ช่วยศาสตราจารย์ ดร. อรรถสิทธิ์ สุรฤกษ์

คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย อนุมัติให้นับวิทยานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญามหาบัณฑิต

..... คณบดีคณะวิศวกรรมศาสตร์
(รองศาสตราจารย์ ดร.บุญสม เลิศหิรัญวงศ์)

คณะกรรมการสอบวิทยานิพนธ์

..... ประธานกรรมการ
(ผู้ช่วยศาสตราจารย์ ดร. พิษณุ คนองชัยยศ)

..... อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก
(ผู้ช่วยศาสตราจารย์ ดร. อรรถสิทธิ์ สุรฤกษ์)

..... กรรมการภายนอกมหาวิทยาลัย
(ผู้ช่วยศาสตราจารย์ ดร. อานนท์ รุ่งสว่าง)

อนรรฆพล เวียงพล : การปรับปรุงการเข้ารหัสเลขคณิตด้วยการจัดกลุ่มความน่าจะเป็นสำหรับเอกสารภาษาอังกฤษ. (AN IMPROVEMENT OF ARITHMETIC CODING USING PROBABILITY CLUSTERING FOR ENGLISH DOCUMENT) อ. ที่ปรึกษาวิทยานิพนธ์หลัก : ผู้ช่วยศาสตราจารย์ ดร.อรรถสิทธิ์ สุรฤกษ์, 69 หน้า.

การเข้ารหัสเลขคณิต (arithmetic coding) เป็นการเข้ารหัสแบบพิจารณาบริบทที่มีประสิทธิภาพด้านค่าอัตราส่วนการบีบอัด (compression ratio: CR) สูงวิธีหนึ่ง อีกทั้งยังเป็นการเข้ารหัสหรือการบีบอัดที่ไม่มีการสูญเสีย (lossless compression) จึงได้รับความนิยมในการบีบอัดไฟล์ข้อมูลที่เป็นทั้งตัวหนังสือ และรูปภาพ การปรับปรุงค่าอัตราส่วนการบีบอัดได้มีการค้นคว้าและวิจัยด้วยวิธีการต่างๆ เช่น การแปลงข้อมูลนำเข้าให้อยู่ในรูปที่เหมาะสม (preprocessing transformation) การประมาณค่าความน่าจะเป็นของสัญลักษณ์ด้วยฟังก์ชันการกระจายของเกาส์ การใช้ต้นแบบที่ปรับเปลี่ยนค่าความน่าจะเป็นได้ และการใช้ทฤษฎีของเบย์ เป็นต้น ซึ่งค่าความน่าจะเป็นเริ่มต้นที่ได้ยังไม่มีความเหมาะสม สามารถปรับปรุงค่าความน่าจะเป็นเริ่มต้นนั้น เพื่อเพิ่มค่าความสามารถในการบีบอัดได้

งานวิจัยนี้ได้นำเสนอเทคนิคในการเพิ่มค่าอัตราส่วนการบีบอัดของการเข้ารหัสเลขคณิตส่วนเพิ่มขึ้นที่สามารถปรับเปลี่ยนค่าความน่าจะเป็นได้ (Incremental Adaptive Arithmetic Coding: IAAC) ในหลายเทคนิคด้วยกัน คือ การประมาณค่าความน่าจะเป็นเริ่มต้นของสัญลักษณ์ด้วยการจัดกลุ่มความน่าจะเป็น (probability clustering) การประมาณค่าความน่าจะเป็นเริ่มต้นของสัญลักษณ์ด้วยการกระจายแบบไวบูลล์ (Weibull distribution) การลดความแตกต่างของค่าความน่าจะเป็นของสัญลักษณ์ (gap reducing) ที่มีความถี่สูงสุดและความถี่ต่ำสุดให้น้อยลง และการลดทอนสัญลักษณ์ที่ไม่ปรากฏ (elimination of unused symbols) รวมถึงการพิจารณาการแบ่งไฟล์ออกเป็นส่วนๆ (file partitioning) เมื่อพบว่าค่าอัตราส่วนการบีบอัดในขณะนั้นมีค่าคงที่หรือมีค่าลดลง

ผลการทดลองเมื่อนำเทคนิคที่นำเสนอไปเข้ารหัสกับไฟล์ในคลังข้อมูลแคลการีคลังข้อมูลแคนเทอเบอร์รีขนาดใหญ่ และไฟล์นิทานภาษาอังกฤษนั้น ให้ผลการปรับปรุงค่าอัตราส่วนการบีบอัดที่ดีขึ้นที่ดีที่สุดสำหรับคลังข้อมูลแคลการีเพิ่มขึ้นถึง 3.8404% และไฟล์นิทานภาษาอังกฤษเพิ่มขึ้นถึง 1.2840% เมื่อเทียบกับการเข้ารหัสเลขคณิตส่วนเพิ่มขึ้นที่สามารถปรับเปลี่ยนค่าความน่าจะเป็นได้ที่กำหนดให้ความน่าจะเป็นเริ่มต้นของแต่ละสัญลักษณ์เท่ากัน

ภาควิชา วิศวกรรมคอมพิวเตอร์ ลายมือชื่อนิสิต อนรรฆพล เวียงพล

สาขาวิชา วิศวกรรมคอมพิวเตอร์ ลายมือชื่อ อ.ที่ปรึกษาวิทยานิพนธ์หลัก Athawat Surapl2

ปีการศึกษา 2553

5070708621 : MAJOR COMPUTER ENGINEERING
KEYWORDS : ARITHMETIC CODING/ LOSSLESS COMPRESSION/ COMPRESSION RATIO/ PROBABILITY ESTIMATION

ANAKAPON WIENGPON : AN IMPROVEMENT OF ARITHMETIC CODING USING PROBABILITY CLUSTERING FOR ENGLISH DOCUMENT THESIS. ADVISOR : ATHASIT SURARERKS, Ph.D., 69 pp.

Arithmetic coding, a very effective lossless encoding algorithm, which basically rely on each probability of symbol. This approach achieves a good compression ratio (CR) as compare to other approaches. Recently, many researchers have studied and proposed various techniques in order to increase the compression ratio such as the preprocessing transformation the input with strings substitution algorithms, m-ary transformation, the estimation of probability of symbol with a specific distribution, adaptive model and Bayes theorem. Such those estimated probability of symbols may improper and thus can be improved in order to gain an increment of compression ratio.

In this thesis, the compression ratio improvement techniques of incremental adaptive arithmetic coding are proposed. First, the probability of symbol clustering and Weibull distribution are employed to generate initial probability of symbols. Second, the gap reducing technique which intends to reduce the gap between the highest and the lowest probability of symbols, the elimination of unused symbols and the file partition techniques.

The experiments were conducted to study and to evaluate the effectiveness of the proposed techniques on standard corpus (e.g. Calgary and Canterbury). Ordinary English novels are also included in the experiment. From the results, the proposed techniques have a good performance as compare to incremental adaptive arithmetic coding with the equally initial probability of symbols. The maximum compression ratio increment was noticed in Calgary corpus at 3.8404% and in English novels at 1.2840%

Department : Computer Engineering Student's Signature อนากพล เวียงพล
Field of Study: Computer Engineering Advisor's Signature Athasit Surarerks
Academic Year : 2010

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงไปได้ด้วยความช่วยเหลืออย่างดียิ่งของอาจารย์ที่ปรึกษา ผู้ช่วยศาสตราจารย์ ดร. อรรถสิทธิ์ สุรฤกษ์ ผู้วิจัยขอกราบขอบพระคุณอาจารย์เป็นอย่างสูงที่ได้ให้คำปรึกษา คำแนะนำ ข้อคิดเห็น และช่วยเหลือแก้ไขข้อบกพร่องต่างๆ จนกระทั่งวิทยานิพนธ์ฉบับนี้เสร็จสมบูรณ์ รวมทั้งกราบขอบพระคุณ ผู้ช่วยศาสตราจารย์ ดร.พิษณุ คนองชัยยศ ในฐานะประธานกรรมการสอบวิทยานิพนธ์ และผู้ช่วยศาสตราจารย์ ดร. อานนท์ รุ่งสว่าง ในฐานะกรรมการสอบวิทยานิพนธ์ ที่กรุณาตรวจแก้วิทยานิพนธ์ฉบับนี้ให้สมบูรณ์ยิ่งขึ้น

ผู้วิจัยใคร่ขอกราบขอบพระคุณบิดา มารดา ที่ส่งเสริมสนับสนุนการศึกษาในทุกๆ ด้านและเป็นกำลังใจอย่างดียิ่งแก่ผู้วิจัย คณาจารย์ภาควิชาวิศวกรรมคอมพิวเตอร์ทุกท่านสำหรับการถ่ายทอดความรู้ และ การถ่ายทอดข้อคิดอันเป็นประโยชน์แก่ผู้วิจัย ตลอดจนขอขอบคุณเพื่อ

สุดท้ายนี้ ผู้วิจัยหวังเป็นอย่างยิ่งว่า งานวิจัยนี้จะเป็นประโยชน์ต่อผู้ที่สนใจหรือผู้ที่เกี่ยวข้อง และหากมีข้อผิดพลาดประการใด ผู้วิจัยขออภัยมา ณ ที่นี่ ด้วย

สารบัญ

สารบัญตาราง

ตารางที่ หน้า

สารบัญภาพ

บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

การติดต่อสื่อสารระหว่างกันในโลกยุคปัจจุบัน นอกจากต้องการได้รับข้อมูลได้อย่างถูกต้องและครบถ้วนแล้ว ประเด็นในเรื่องประสิทธิภาพของการรับส่ง (transmission efficiency) เป็นสิ่งหนึ่งที่ได้มีการวิจัยและปรับปรุงเรื่อยมา การวัดประสิทธิภาพของการรับส่งของข้อมูลด้านหนึ่งที่ได้รับความน่าสนใจจากงานวิจัยหลายงานเรื่อยมา [1 - 6] คือการวัดค่าอัตราส่วนการบีบอัด (compression ratio : CR) โดยค่าอัตราส่วนมีค่ามากถือว่ามีประสิทธิภาพในการรับส่งสูง กล่าวอีกนัยหนึ่งคือค่าอัตราส่วนการบีบอัดมีค่ามาก ข้อมูลจะมีขนาดเล็กลง ซึ่งเป็นผลทำให้เราสามารถที่จะได้รับหรือส่งข้อมูลนั้นได้รวดเร็วและมีประสิทธิภาพมากขึ้นเมื่อเทียบกับข้อมูลที่มีค่าอัตราส่วนการบีบอัดที่มีค่าน้อยกว่า การทำให้ข้อมูลมีขนาดเล็กลงเพื่อประโยชน์ในการส่งอย่างมีประสิทธิภาพนั้นแต่เพียงอย่างเดียวนั้นจะไม่มีประโยชน์ หากขาดการทำให้ข้อมูลที่บีบอัดไปแล้วนั้นกลับคืนสู่ผู้รับเหมือนต้นฉบับได้ เราสามารถเรียกกระบวนการทั้งสองที่มีกระบวนการในการบีบอัดข้อมูล (compression process) และกระบวนการในการคลายตัวข้อมูล (decompression process) ว่ากระบวนการเข้ารหัสและถอดรหัส (encoding and decoding process) นอกจากนี้เพื่อความสะดวกและง่ายต่อการเข้าใจเราจึงนิยมเรียกกระบวนการทั้งสองว่า การเข้ารหัส (coding process)

กระบวนการเข้ารหัสนี้สามารถแยกได้เป็นสองประเภทตามจุดประสงค์ในการใช้งาน คือแบบยอมให้เสียคุณภาพไปบางส่วน (lossy compression) และแบบคงสภาพเหมือนต้นฉบับ (lossless compression) [7] โดยที่แบบยอมให้เสียคุณภาพไปบางส่วนนั้นจะยอมให้มีการสูญเสียคุณภาพของข้อมูลภายหลังการถอดรหัสแล้วลงไปบ้าง ผู้รับสารส่วนใหญ่จะไม่สามารถแยกความแตกต่างออกจากต้นฉบับเดิมได้ จึงนิยมใช้ประเภทนี้สำหรับข้อมูลจำพวก รูปภาพ เสียง หรือ วีดีโอ ส่วนการบีบอัดแบบคงสภาพเหมือนต้นฉบับนั้น ผู้รับสารจะได้ข้อมูลถูกต้องครบถ้วนเหมือนต้นฉบับภายหลังจากการถอดรหัส

นอกจากนี้เรายังสามารถแบ่งกระบวนการหรืออัลกอริทึมสำหรับการเข้ารหัส ตามลักษณะของวิธีที่ใช้ โดยประกอบไปด้วยสองประเภทคือ การเข้ารหัสโดยการพิจารณาบริบท (context-based) และการเข้ารหัสโดยอาศัยพจนานุกรม (dictionary-based) โดยการเข้ารหัสโดยการพิจารณาบริบทนั้น อาศัยหลักการของการกำหนดให้บิตคำตอบ มีความยาวสั้นเมื่อข้อมูลหรือสัญลักษณ์นำเข้านั้นมีความถี่หรือความน่าจะเป็นในการใช้งานมาก ส่วนสัญลักษณ์ที่มีความถี่หรือความน่าจะเป็นน้อยจะให้บิตคำตอบที่ยาวกว่าได้ การเข้ารหัสด้วยวิธีนี้มีอยู่หลายวิธีด้วยกันได้แก่วิธีการของ บูโรวส์-วีลเลอร์ (Burrows-Wheeler

compression algorithm - BWCA) [8] การทำนายโดยการจับคู่เพียงบางส่วน (prediction by partial matching - PPM) [9] การเข้ารหัสของฮัฟฟ์แมน (Huffman coding) [10] การเข้ารหัสของโกลอมบ์และไรซ์ (Golomb-Rice coding) [6, 11] และการเข้ารหัสเลขคณิต (arithmetic coding) [12] เป็นต้น ส่วนแบบแผนอีกด้านหนึ่งคือการเข้ารหัสโดยอาศัยพจนานุกรมคืออัลกอริทึมของการเข้ารหัสของลิมเพล-ซิฟ และคณะ (Lempel-Ziv et. al.) [13] โดยเป็นการเข้ารหัสข้อมูล โดยการเก็บแต่ค่าดัชนี (index) ของสัญลักษณ์แทนข้อมูลนำเข้านั้น ซึ่งดัชนีเหล่านี้อาจแทนข้อมูลนำเข้าในหลายๆ ระดับ เช่น ในระดับแต่ละสัญลักษณ์ (alphabet) หรือในระดับกลุ่มของสัญลักษณ์ เช่น ระดับสองหรือสามสัญลักษณ์ หรือระดับคำ โดยที่เราจะเรียกหน่วยความจำที่เก็บดัชนีที่เป็นตัวแทนของสัญลักษณ์หรือกลุ่มของสัญลักษณ์เหล่านั้นว่า พจนานุกรม (dictionary) ซึ่งพจนานุกรมนี้มีความสำคัญอย่างมากต่อกระบวนการถอดรหัส ในการศึกษาของเมอร์ทีและคณะ (Murthy C. et. al.) [4] ได้ศึกษาถึงวิธีการลดขนาดของพจนานุกรมเพื่อเพิ่มประสิทธิภาพในการบีบอัดด้วย นั่นแสดงให้เห็นว่าพจนานุกรมที่มีขนาดใหญ่จะลดทอนประสิทธิภาพในการรับส่งได้ โดยเฉพาะอย่างยิ่งค่าอัตราส่วนการบีบอัดจะมีค่าลดลงตามไปด้วย การศึกษาเพื่อหาวิธีการเก็บพจนานุกรมให้มีขนาดเล็กและมีประสิทธิภาพ ถือเป็นงานวิจัยด้านหนึ่งที่มีความสำคัญและได้รับความสนใจเรื่อยมา

งานวิจัยหลายงานวิจัย [14 - 16] ระบุว่าการเข้ารหัสเลขคณิตนั้นให้ค่าอัตราส่วนการบีบอัดที่สูงวิธีหนึ่ง ซึ่งในบางครั้งค่าเอนโทรปียังสามารถเข้าใกล้ค่าเอนโทรปีที่ดีที่สุดได้ (optimal entropy) [14] อีกทั้งเป็นการเข้ารหัสที่ไม่อาศัยพจนานุกรม โดยการทำงานของการเข้ารหัสเลขคณิต ใช้การคำนวณที่ต้องการค่าความแม่นยำในระดับสูง [7, 14] เพื่อให้คำตอบที่ต้องการอยู่ในรูปของจำนวนจริงที่อยู่ในช่วงๆ หนึ่ง โดยอาศัยค่าความน่าจะเป็นของแต่ละสัญลักษณ์ นอกจากนั้นยังต้องอ่านสัญลักษณ์ของไฟล์นำเข้าจนครบทุกสัญลักษณ์จึงจะสามารถผลิตคำตอบนั้นได้ ทำให้การเข้ารหัสเลขคณิตเป็นการเข้ารหัสที่ค่อนข้างนานกว่าที่จะได้คำตอบออกมา ถึงแม้ว่าจะมีแนวทางในการปรับปรุงอัลกอริทึมดังกล่าวให้ใช้เวลาเข้ารหัสน้อยลง ด้วยการลดความซับซ้อนในการคูณแล้วก็ตาม แต่ในงานวิจัยนี้มุ่งเน้นศึกษาถึงการเพิ่มค่าอัตราส่วนการบีบอัดเป็นสำคัญ ดังนั้นประเด็นทางด้านเวลาในการเข้ารหัส [17] จึงเป็นเรื่องที่สำคัญรองลงไป

แนวทางในการปรับปรุงอัลกอริทึมของการเข้ารหัสเลขคณิตนั้น มีอยู่ด้วยกันหลายแนวทางเช่น การปรับปรุงอัลกอริทึมของการเข้ารหัสเอง [7, 14, 18 - 19] ในงานวิจัยของโฮวาร์ดและคณะ (Howard P. G. et. al.) [14] ได้นำเสนอการเข้ารหัสเลขคณิตส่วนเพิ่มขึ้น (incremental arithmetic coding) ซึ่งเป็นการเข้ารหัสเลขคณิตแบบหนึ่ง โดยการเข้ารหัสเลขคณิตส่วนเพิ่มขึ้นนี้สามารถแก้ข้อด้อยของการเข้ารหัสเลขคณิตแบบดั้งเดิมได้หลายประการ โดยมีหลักการโดยย่อคือ เมื่อรู้ว่ามีบิตของคำตอบร่วมกัน (common bits) ของช่วง

ต่ำสุด (low) และ ช่วงสูงสุด (high) ให้ผลิตบิตคำตอบนั้นทันที จากนั้นให้ขยายช่วงต่ำสุดและช่วงสูงสุดออกเป็นสองเท่า นอกจากนี้งานวิจัยทางด้านการเตรียมข้อมูลให้อยู่ในโครงสร้างที่เหมาะสม [1, 15, 20 - 22] ดังเช่นจากงานวิจัยของจุง (Hyoung Joong K.) [21 - 22] ได้นำเสนอในเรื่องการแปลงลำดับของเลขฐานสองที่มีความยาวมาก ให้กลายเป็นลำดับของ m-ary ที่สั้นกว่า ก่อนไปทำการเข้ารหัสเลขคณิต ซึ่งวิธีการนี้จะใช้เวลาในการเข้ารหัสที่ไม่นับเวลาที่ใช้ในการแปลงลำดับนั้นน้อยลงตามไปด้วย การปรับปรุงวิธีในการประมาณค่าความน่าจะเป็นของแต่ละสัญลักษณ์ [2, 5, 23 - 26] ก็ได้รับความนิยมจากนักวิจัยหลายท่านด้วยกัน ดังจะเห็นได้จากงานวิจัยของโพเวล (M. Powell) [5] ที่ได้รายงานไว้ว่าการประมาณค่าความน่าจะเป็นของแต่ละสัญลักษณ์ สามารถทำได้โดยการปรับเปลี่ยนค่าความน่าจะเป็นไปในระหว่างกระบวนการเข้ารหัส หรือ ต้นแบบที่สามารถปรับเปลี่ยนค่าความน่าจะเป็นได้ (adaptive model) โดยต้นแบบนี้จะส่งผลให้ค่าอัตราส่วนการบีบอัดที่สูงกว่า เมื่อไม่สามารถปรับเปลี่ยนค่าความน่าจะเป็น (fixed model) นอกจากนี้ยังมีการประมาณค่าความน่าจะเป็นในระดับกลุ่มของสัญลักษณ์ ซึ่งนิยมใช้การเข้ารหัสที่อาศัยพจนานุกรมมากกว่า [4, 27] การประมาณค่าความน่าจะเป็นโดยฟังก์ชันความหนาแน่นของความน่าจะเป็น (Probability density function: PDF) แบบลาปลาซ (Laplacian distribution) [28] และแบบเกาส์ (Gaussian distribution) [24, 29] เป็นต้น

งานวิจัยนี้จึงขอนำเสนอแนวทางในการปรับปรุงอัลกอริทึมของการเข้ารหัสเลขคณิตส่วนเพิ่มขึ้นเพื่อให้ได้ผลของค่าอัตราส่วนการบีบอัดที่สูงขึ้นออกเป็นสองวิธี โดยวิธีแรกอาศัยการพิจารณาเรื่องการประมาณค่าความน่าจะเป็นเริ่มต้นของสัญลักษณ์ด้วยการจัดกลุ่มความน่าจะเป็น (probability clustering) อีกวิธีหนึ่งคือการประมาณค่าความน่าจะเป็นเริ่มต้นด้วยการกระจายแบบไวบูลล์ (Weibull distribution) โดยทั้งสองวิธีนี้จะทำงานอยู่บนเทคนิคของการเข้ารหัสและถอดรหัสเลขคณิตส่วนเพิ่มขึ้นที่สามารถปรับเปลี่ยนค่าความน่าจะเป็นได้ อีกทั้งได้เพิ่มเติมเทคนิคอื่นๆ เพื่อเพิ่มค่าอัตราส่วนการบีบอัดได้แก่ การลดความแตกต่างของค่าความน่าจะเป็นของสัญลักษณ์ (Gap reducing technique) ที่มีความถี่สูงสุดและความถี่ต่ำสุดให้แตกต่างน้อยลงไป และการลดทอนสัญลักษณ์ที่ไม่ปรากฏ (Elimination of unused symbols technique) รวมถึงการพิจารณาการแบ่งไฟล์ออกเป็นส่วนๆ (File partitioning technique) เมื่อพบว่าค่าอัตราส่วนการบีบอัดในขณะนั้นไม่ได้ดีขึ้นหรือมีค่าลดลง

1.2 วัตถุประสงค์ของการวิจัย

เพื่อปรับปรุงประสิทธิภาพในการบีบอัดของเอกสารภาษาอังกฤษ เพื่อให้ค่าอัตราส่วนการบีบอัดมีค่าสูงขึ้นของการเข้ารหัสเลขคณิตส่วนเพิ่มขึ้นที่สามารถปรับเปลี่ยนค่าความน่าจะเป็นได้

1.3 ขอบเขตของการวิจัย

1.3.1. พิจารณาเฉพาะการเข้ารหัสเลขคณิตส่วนเพิ่มขึ้นที่สามารถปรับเปลี่ยนค่าความน่าจะเป็นได้ ในระดับสัญลักษณ์แอสกี (ASCII Symbol) จำนวน 256 สัญลักษณ์

1.3.2. ใช้คลังข้อมูลแคลการี คลังข้อมูลแคนเทอเบอร์รีขนาดใหญ่ และนิทานภาษาอังกฤษ เพื่อเป็นต้นแบบในการศึกษาการวิเคราะห์กลุ่มความถี่ของสัญลักษณ์ เพื่อวิเคราะห์หาฟังก์ชันความหนาแน่นของความน่าจะเป็นที่เหมาะสมที่สุด (Fit distribution)

1.3.3. นำเสนอเฉพาะการเปรียบเทียบค่าอัตราส่วนการบีบอัดระหว่าง การเข้ารหัสเลขคณิตส่วนเพิ่มขึ้นที่สามารถปรับเปลี่ยนค่าความน่าจะเป็นได้ การปรับปรุงการเข้ารหัสเลขคณิตส่วนเพิ่มขึ้นที่ปรับปรุงโดยการประมาณค่าความน่าจะเป็นเริ่มต้นของสัญลักษณ์ด้วยการจัดกลุ่มความน่าจะเป็น และการประมาณค่าความน่าจะเป็นเริ่มต้นของสัญลักษณ์ด้วยการกระจายแบบไวบูลล์

1.3.4. ใช้เทคนิคเพิ่มเติมเพื่อช่วยเพิ่มค่าอัตราส่วนการบีบอัด ได้แก่ การลดความแตกต่างของค่าความน่าจะเป็นของสัญลักษณ์ ที่มีความถี่สูงสุดและความถี่ต่ำสุดให้น้อยลง การลดทอนสัญลักษณ์ที่ไม่ปรากฏ และ การพิจารณาการแบ่งไฟล์ออกเป็นส่วนๆ เมื่อพบว่าค่าอัตราส่วนการบีบอัดในขณะนั้นมีค่าคงที่หรือมีค่าลดลง

1.4 ประโยชน์ที่คาดว่าจะได้รับ

ได้วิธีการเพิ่มค่าอัตราส่วนการบีบอัดของการเข้ารหัสเลขคณิตส่วนเพิ่มขึ้นที่สามารถปรับเปลี่ยนค่าความน่าจะเป็นได้ ของเอกสารประเภทข้อความภาษาอังกฤษ

1.5 วิธีดำเนินการวิจัย

1.5.1. ศึกษางานวิจัยและข้อมูลเอกสารที่เกี่ยวข้องกับการเข้ารหัสเลขคณิตแบบต่างๆ ตลอดจนวิธีการจัดกลุ่มแบบต่างๆ

1.5.2. วิเคราะห์หลักการการเข้ารหัสเลขคณิตส่วนเพิ่มขึ้นที่สามารถปรับเปลี่ยนค่าความน่าจะเป็นได้ ใช้เวลาออกแบบอัลกอริทึมสำหรับการเตรียมค่าความน่าจะเป็นเริ่มต้น และเทคนิคเพิ่มเติมต่างๆ

1.5.3. ทดสอบค่าอัตราส่วนการบีบอัดของงานวิจัยที่ได้ออกแบบไว้ เปรียบเทียบกับการเข้ารหัสเลขคณิตส่วนเพิ่มขึ้นที่สามารถปรับเปลี่ยนค่าความน่าจะเป็นได้

1.5.4. สรุปผลการวิจัยและตีพิมพ์ผลงานวิจัย

1.5.5. เรียบเรียงและจัดทำวิทยานิพนธ์

1.6 ผลงานที่ตีพิมพ์จากวิทยานิพนธ์

ส่วนหนึ่งของวิทยานิพนธ์นี้ได้รับการตีพิมพ์เป็นบทความทางวิชาการในหัวข้อเรื่องดังนี้

1.6.1. A. Wiengpon and A. Surarerks, **"A Modified Version of Adaptive Arithmetic Encoding Algorithm".** In Proceeding of the 14th International Annual Symposium on Computational Science and Engineering: ANSCSE14, Chiang Rai, Thailand, March 23-26, 2010.

1.6.2. A. Wiengpon and A. Surarerks, **"Weibull based Incremental Adaptive Arithmetic Coding with File Partition Technique"**. 10th International Symposium on Communications and Information Technology, ISCIT 2010, Tokyo, Japan, October 26 - 29, 2010.

บทที่ 2
เอกสารและงานวิจัยที่เกี่ยวข้อง

2.1 ความรู้พื้นฐานและทฤษฎีที่เกี่ยวข้อง

2.1.1 นิยามเบื้องต้น

ในหัวข้อนี้จะกล่าวถึงนิยามและสัญลักษณ์ที่เกี่ยวข้อง

นิยามที่ 2.1 *เซตอักขระ และสัญลักษณ์ (Alphabet and Symbol) เซตอักขระ* (A) *คือเซตจำกัดของสัญลักษณ์ โดยจำนวนสมาชิกของเซตอักขระเขียนแทนด้วย* $|A|$ *และสมาชิกแต่ละตัวในเซตอักขระถูกเรียกว่าสัญลักษณ์ โดยในที่นี้สัญลักษณ์จะหมายถึงสัญลักษณ์แอสกีจำนวน 256 สัญลักษณ์*

นิยามที่ 2.2 *ลำดับ (Sequence) คือ สมาชิกในเซตอักขระ A ที่เรียงต่อกัน*

นิยามที่ 2.3 *แบบจำลอง (Model) ถ้าให้ A เป็นเซตของอักขระ แบบจำลอง M นิยามดังนี้*

$$\begin{aligned} M: A \rightarrow \quad & [0,1) \\ M: a_i \rightarrow \quad & P_M(a_i) \end{aligned} \tag{1}$$

โดยแบบจำลองดังกล่าวเป็นการจับคู่ระหว่างความน่าจะเป็นของสัญลักษณ์แต่ละสัญลักษณ์ $(P_m(a_i))$

จากงานวิจัยแชนนอน [30] นับเป็นงานวิจัยเริ่มต้นที่ก่อตั้งทฤษฎีของการบีบอัดข้อมูล (theory of data compression) ได้นำเสนอแบบจำลองลำดับการใช้สัญลักษณ์สำหรับการเข้ารหัสไว้ โดยใช้สัญลักษณ์ในภาษาอังกฤษที่ได้นิยามไว้เพียง 27 สัญลักษณ์ คือ $A = \{a, b, .. z\} \cup \{\Delta\}$ โดยที่ Δ แทนเว้นวรรค (spacebar) งานวิจัยนี้ได้แบ่งประเภทของลำดับในสัญลักษณ์ในภาษาอังกฤษไว้เป็นหกแบบได้แก่

- แบบจำลองอันดับศูนย์ (zero-order model) หมายถึงสัญลักษณ์ในภาษาที่แต่ละสัญลักษณ์มีความเป็นอิสระต่อกัน
- แบบจำลองอันดับหนึ่ง (first-order model) หมายถึงสัญลักษณ์ในภาษาที่แต่ละสัญลักษณ์มีความเป็นอิสระต่อกันและลำดับนี้จะมีค่าความน่าจะ

เป็นของแต่ละสัญลักษณ์เข้ามาเกี่ยวข้อง ตัวอย่างเช่นสัญลักษณ์ ‘a’, ‘e’ จะเกิดขึ้นด้วยความน่าจะเป็นมากกว่าสัญลักษณ์ ‘q’

- แบบจำลองอันดับสอง (second-order model) หรือเรียกว่า ไดแกรม (digram) หมายถึงสัญลักษณ์ในภาษาที่แต่ละสัญลักษณ์ขึ้นอยู่กับสัญลักษณ์ก่อนหน้าหนึ่งสัญลักษณ์ ตัวอย่างเช่นสัญลักษณ์ ‘u’ จะมีค่าความน่าจะเป็นที่สูงกว่าเมื่อสัญลักษณ์ข้างหน้าเป็น ‘q’ เมื่อเทียบกับสัญลักษณ์ก่อนหน้าสัญลักษณ์อื่น
- แบบจำลองอันดับสาม (third-order model) หรือเรียกว่า ไตรแกรม (trigram) หมายถึงสัญลักษณ์ในภาษาที่แต่ละสัญลักษณ์ขึ้นอยู่กับสัญลักษณ์ก่อนหน้าสองสัญลักษณ์
- แบบจำลองอันดับหนึ่งในระดับคำ (first-order word model) หมายถึงสัญลักษณ์ในระดับคำในภาษาที่แต่ละคำมีความเป็นอิสระต่อกัน ค่าความน่าจะเป็นตามความเหมาะสม
- แบบจำลองอันดับสองในระดับคำ (second-order word model) หมายถึงสัญลักษณ์ในระดับคำในภาษาที่แต่ละคำมีความเป็นอิสระต่อกัน ค่าความน่าจะเป็นนั้นถูกต้อง

นิยาม 2.4 *เอนโทรปี (Entropy) เป็นค่าที่บ่งบอกถึงปริมาณของสาระโดยเฉลี่ยที่มีอยู่ข้อมูลเหล่านั้น โดยพิจารณาจากค่าความน่าจะเป็นของแต่ละสัญลักษณ์* $P(x_i)$ *ค่าเอนโทรปีสามารถคำนวณได้จาก*

$$H = \sum_{i=1}^{n} P(x_i) lg \frac{1}{P(x_i)} \quad (2)$$

โดยที่

x_i *คือสัญลักษณ์ที่* i *ในข้อมูลจากสัญลักษณ์* n *สัญลักษณ์*

H *คือ เอนโทรปี*

หน่วยของเอนโทรปีคือ บิต/สัญลักษณ์ (bit/symbol) เนื่องจากสูตรของเอนโทรปีอ้างถึงค่าความน่าจะเป็นแบบอิงความถี่สัมพัทธ์ ไม่ใช่ความถี่ที่แท้จริง นอกจากในงานวิจัยของอีริค (Eric B.) [7] ยังระบุว่าค่า

$$\sum_{i=1}^{n} n(x_i) lg \frac{1}{P(x_i)} \quad (3)$$

คือค่าจำนวนบิตที่ดีที่สุดของบิตคำตอบรวมในกระบวนการเข้ารหัสเลขคณิตอีกด้วย

ตัวอย่างที่ 2.1 การคำนวณค่าเอนโทรปี

$$
\begin{aligned}
P_{M_1}(x) &= \{0.2, 0.2, 0.2, 0.2, 0.2\} \\
H_{M_1} &= 5 \times 0.2 \times lg5 \\
&= 2.32
\end{aligned}
$$

$$
\begin{aligned}
P_{M_2}(x) &= \{0.25, 0.25, 0.25, 0.125, 0.125\} \\
H_{M_2} &= 3 \times 0.25 \times lg4 + 2 \times 0.125 \times lg8 \\
&= 1.5 + 0.75 \\
&= 2.25
\end{aligned}
$$

$$
\begin{aligned}
P_{M_3}(x) &= \{0.5, 0.125, 0.125, 0.125, 0.125\} \\
H_{M_3} &= 0.5 \times lg2 + 4 \times 0.125 \times lg8 \\
&= 0.5 + 1.5 \\
&= 2
\end{aligned}
$$

$$
\begin{aligned}
P_{M_4}(x) &= \{0.75, 0.0625, 0.0625, 0.0625, 0.0625\} \\
H_{M_4} &= 0.75 \times lg\frac{4}{3} + 4 \times 0.0625 \times lg16 \\
&= 0.3 + 1 \\
&= 1.3
\end{aligned}
$$

□

จากตัวอย่างการคำนวณค่าเอนโทรปีดังข้างต้น เป็นที่น่าสังเกตว่าเมื่อค่าความน่าจะเป็นยิ่งมีค่าแตกต่างกัน หรือมีความเบ้ยิ่งมากนั้น (highly screwed) จะส่งผลให้ค่าเอนโทรปีที่ได้มีค่าลดลงเมื่อเทียบกับค่าความน่าจะเป็นแบบเท่ากัน

เซด (Said A.) [19] ได้กล่าวถึงค่าความน่าจะเป็นที่เป็นค่าโดยประมาณของสัญลักษณ์ในภาษาอังกฤษบางสัญลักษณ์ อีกทั้งยังสรุปว่า จำนวนบิตที่ดีที่สุดนั้น (Optimal number of bits) ของแต่ละสัญลักษณ์ ขึ้นอยู่กับค่าความน่าจะเป็นโดยประมาณนั้น และมีค่าเท่ากับค่าสาระ (information, $-\lg P(x)$) ของสัญลักษณ์นั้น ซึ่งสอดคล้องกับสมการ (3) จำนวนบิตที่ดีที่สุดของสัญลักษณ์ในภาษาอังกฤษบางสัญลักษณ์นั้นแสดงได้ดังตารางที่ 2.1

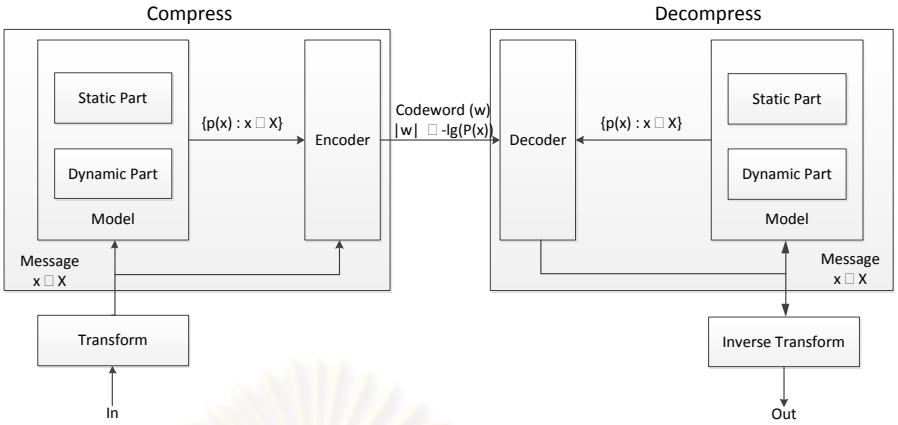
ตารางที่ 2.1 ค่าความน่าจะเป็นโดยประมาณของสัญลักษณ์และเครื่องหมายวรรคตอนบางสัญลักษณ์ในภาษาอังกฤษที่เรียงลำดับตามสัญลักษณ์แอสกี

สัญลักษณ์ (x)	สัญลักษณ์แอสกี (ASCII Symbol)	ค่าความน่าจะเป็น $P(x)$	จำนวนบิตที่ดีที่สุด (Optimal number of bits = $-\lg P(x)$)
Space	32	0.1524	2.714
,	44	0.0136	6.205
.	46	0.0056	7.492
A	65	0.0017	9.223
B	66	0.0009	10.065
C	67	0.0013	9.548
a	97	0.0595	4.071
b	98	0.0119	6.391
c	99	0.0230	5.441
d	100	0.0338	4.887
e	101	0.1033	3.275
f	102	0.0227	5.463
t	116	0.0707	3.823
z	122	0.0005	11.069

จากตารางที่ 2.1 พบว่าเมื่อสัญลักษณ์เป็น ‘Space” แล้วจะให้จำนวนบิตที่ดีที่สุดเป็น 3 บิต ซึ่งต่างจากเมื่อสัญลักษณ์เป็น ‘z’ ที่มีความน่าจะเป็นในการใช้น้อยกว่ามาก และต้องใช้จำนวนบิตที่ดีที่สุดถึง 12 บิตโดยประมาณ

2.1.2 ต้นแบบในการเข้ารหัสโดยการพิจารณาบริบท

การเข้ารหัสเลขคณิตจัดอยู่ในการเข้ารหัสโดยการพิจารณาบริบทที่คงสภาพเหมือนต้นฉบับ ดังนั้นการทำความเข้าใจต้นแบบของการเข้ารหัสด้วยวิธีนี้ จึงเป็นพื้นฐานที่สำคัญประกอบความเข้าใจในการศึกษาต่อไป โดยเราสามารถเขียนต้นแบบการเข้ารหัสนั้น ให้อยู่ในรูปง่ายต่อการเข้าใจ โดยองค์ประกอบหลักๆ ตลอดจนขั้นตอนที่สำคัญซึ่งได้แสดงไว้ดังรูปที่ 2.1 โดยมีสองส่วนได้แก่ ส่วนที่ใช้ในการเข้ารหัส หรือการบีบอัด (compressor) และส่วนที่ใช้ในการถอดรหัสหรือการคลายตัว (decompressor)



รูปที่ 2.1 ต้นแบบในการเข้ารหัสและการถอดรหัส

กระบวนการเข้ารหัส เริ่มต้นจากเมื่อข้อมูลนำเข้า (input) ซึ่งอาจจะเป็นระดับสัญลักษณ์ ระดับคำ หรือระดับอื่นๆ มาผ่านการแปลงข้อมูลให้อยู่ในรูปที่เหมาะสมหรือการเตรียมข้อมูลก่อนการเข้ารหัส (transform: preprocessing the input) จากนั้นเมื่อเข้าสู่กระบวนการเข้ารหัสแล้ว สามารถเลือกต้นแบบในการประมาณค่าความน่าจะเป็นได้สองแบบ คือไม่สามารถปรับเปลี่ยนค่าความน่าจะเป็น (static part, fixed model) และแบบที่มีการปรับเปลี่ยนค่าความน่าจะเป็นได้ (dynamic part, adaptive model) จากนั้นเมื่อพิจาณาได้ค่าความน่าจะเป็นของข้อมูลนำเข้าตัวนั้นแล้ว จึงส่งทั้งค่าความน่าจะเป็นตลอดจนข้อมูลนำเข้าตัวนั้นไปเข้ารหัสที่ตัวเข้ารหัส (Coder) เพื่อผลิตคำตอบ (codeword: w) ต่อไป โดยที่ค่าขนาดของคำตอบ (|w|) ที่ได้นั้น จะเป็นไปตามหรือมีค่าประมาณตามค่าสาระ ซึ่งมีค่าเท่ากับ $-\lg \mathrm{P}(\mathrm{x})$ ส่วนกระบวนการถอดรหัสนั้นสามารถพิจารณาได้เช่นเดียวกันกับกระบวนการเข้ารหัส เพียงแต่เริ่มต้นการถอดรหัสด้วยคำตอบหรือ codeword ก่อนที่จะไปสิ้นสุดที่การเตรียมข้อมูลหลังการถอดรหัส (inverse transform: post-processing the output) และได้ค่าผลลัพธ์ (output) เหมือนกับข้อมูลนำเข้า

นิยาม 2.5 *ตัวเข้ารหัส (Encoder) คืออัลกอริทึมที่ใช้ในการเข้ารหัสข้อมูลนำเข้าให้กลายเป็นบิตคำตอบ และ ตัวถอดรหัส (Decoder) คืออัลกอริทึมที่ใช้ในการถอดรหัสบิตคำตอบให้กลายเป็นข้อมูลนำเข้า*

นิยาม 2.6 *บิตคำตอบ (codeword) คือรูปแบบแทนผลลัพธ์ของการเข้ารหัส*

2.1.3 การเข้ารหัสเลขคณิต

การเข้ารหัสเลขคณิต [23] อาศัยหลักการของการแบ่งช่วงที่กำหนดลงไปเรื่อยๆ โดยสามารถแสดงผลลัพธ์ได้ด้วยจำนวนจริงที่อยู่ในช่วง [0, 1) ด้วยค่าของความน่าจะเป็นของแต่ละสัญลักษณ์ในเซตของสัญลักษณ์ที่สนใจ (code value representation) โดยผลจากการคำนวณตามอัลกอริทึมที่ 1.1 แล้วเราจะได้ค่าจำนวนจริงที่อยู่ในช่วงหนึ่ง (ค่าสูงสุด-ค่าต่ำสุด) ที่ใช้เป็นบิตคำตอบ อัลกอริทึมในการเข้ารหัสและการถอดรหัสเลขคณิตนี้สามารถแสดงได้ในอัลกอริทึมที่ 1.1 การเข้ารหัสเลขคณิตและอัลกอริทึมที่ 1.2 การถอดรหัสเลขคณิตตามลำดับ

Algorithm 1.1 arithmetic coding

```
1   Input:    File (a1a2..an),
              array of cumulative frequency of ai (cum_freq[ai])
2   Output:   low, high, V
3   Begin
4       Let: low = 0, high = 1, i = 0
5       Do
6           range  = high  – low
7           high = low + range * cum_freq[ai]
8           low = low + range * cum_freq[ai-1]
9           i = i +1
10      While (i<n)
11  return V where low ≤ V < high
12  End
```

ในการถอดรหัสเลขคณิตเราจำเป็นที่จะต้องทราบถึงตำแหน่งสำหรับหยุดการคิดคำนวณ โดยที่เราสามารถทำได้ในหลายลักษณะได้แก่ การส่งจำนวนสัญลักษณ์ทั้งหมดของไฟล์ไปให้ตัวถอดรหัส หรือการกำหนดสัญลักษณ์พิเศษเช่น <EOF> เข้าไปด้วยในการเข้ารหัส เมื่อถึงเวลาถอดรหัส ถ้าพบเงื่อนไขอย่างใดอย่างหนึ่งดังที่ได้กล่าวไปแล้วให้หยุดการเข้ารหัสได้ โดยจากอัลกอริทึมที่ 1.2 นั้นให้หยุดการเข้ารหัสเมื่อเจอสัญลักษณ์ <EOF>

Algorithm 1.2: arithmetic decoding

```
1   Input:    array of cumulative frequency of ai (cum_freq[ai]), V
2   Output    File (a1a2..an)
    :
3   Begin
4       Let: low = 0, high = 1, i = 0, File = Φ
5       Do
6           Find symbol that meet this constraint:
            cum_freq[ai-1] ≤ (V – low)/(high – low) < cum_freq[ai]
7           range  = high  – low
```

```
8               high = low + range * cum_freq[a_i]
9               low = low + range * cum_freq[a_i-1]
10              File += symbol
11              i = i+1
12          While (!EOF)
13      Return File
14      End
```

ตัวอย่างที่ 2.2 การเข้ารหัสและถอดรหัสเลขคณิต

กำหนดให้ค่าความน่าจะเป็นของสัญลักษณ์ {a, b, c} แสดงดังในตารางที่ 2.2 ต้องการเข้ารหัสของไฟล์ข้อมูลที่ประกอบไปด้วยกลุ่มของสัญลักษณ์ bbbc โดยสมมติให้ L แทนค่า low และ H แทนค่า high ขั้นตอนในการเข้ารหัสเลขคณิตแสดงได้ดังในตารางที่ 2.3 และขั้นตอนในการถอดรหัสเลขคณิตแสดงดังในตารางที่ 2.4 ตามลำดับ

ตารางที่ 2.2 ตัวอย่างค่าความน่าจะเป็นของแต่ละสัญลักษณ์

สัญลักษณ์	ค่าความน่าจะเป็น	ช่วง
a	0.4	[0.0, 0.4)
b	0.5	[0.4, 0.9)
c	0.1	[0.9, 1.0)

ตารางที่ 2.3 ตัวอย่างการเข้ารหัสเลขคณิต

ช่วงปัจจุบัน	ช่วงย่อย			สัญลักษณ์
	a	b	c	
[0.0, 1.0)	[0.0, 0.4)	[0.4, 0.9) L: 0 + (1-0)(0.4) = 0.4 H: 0 + (1-0)(0.9) = 0.9	[0.9, 1)	b
[0.4, 0.9)	[0.4, 0.6)	[0.6, 0.85) L: 0.4+(0.9-0.4)(0.4) = 0.6 H: 0.4+(0.9-0.4)(0.9) = 0.85	[0.85, 0.9)	b
[0.6, 0.85)	[0.6, 0.7)	[0.7, 0.825) L: 0.6+(0.85-0.6)(0.4) = 0.7 H: 0.6+(0.85-0.6)(0.9) = 0.825	[0.825, 0.85)	b
[0.7, 0.825)	[0.7,0.75)	[0.75, 0.8125)	[0.8125, 0.825) L: 0.7+(0.825-0.7)(0.9) = 0.8125 H: 0.7+(0.825-0.7)(1) = 0.825	c

สมมติให้เลือกค่า V = 0.8125 ($0.8125 \leq V < 0.825$) เป็นผลลัพธ์ของการเข้ารหัส

ตารางที่ 2.4 ตัวอย่างการถอดรหัสเลขคณิต

V	a	b	c	สัญลักษณ์
0.8125		[0.4, 0.9)		b
(0.8125-0.4)/(0.9-0.4)= 0.825		[0.4, 0.9)		b
(0.825-0.4)/(0.9-0.4) = 0.85		[0.4, 0.9)		b
(0.85-0.4)/(0.9-0.4) = 0.90			[0.9, 1)	c

□

2.1.4 การเข้ารหัสเลขคณิตที่สามารถปรับเปลี่ยนค่าความน่าจะเป็นได้

ในระหว่างกระบวนการเข้ารหัสและถอดรหัสนั้น ค่าความน่าจะเป็นที่ต้องการคือค่าความน่าจะเป็นที่ก่อให้เกิดค่าอัตราส่วนการบีบอัดสูงสุด ในงานวิจัยของโพเวล (Powell M.) [5] พบว่าค่าความน่าจะเป็นที่ปรับเปลี่ยนค่าได้นี้ให้ค่าอัตราส่วนการบีบอัดดีกว่าแบบที่ไม่สามารถปรับเปลี่ยนค่าความน่าจะเป็น ส่วนความน่าจะเป็นที่ดีที่สุดนั้นคือความน่าจะเป็นที่แท้จริง (exact value) ของข้อมูลนั้น แต่เราจะต้องเสียเวลาที่ใช้ในการอ่านไฟล์ข้อมูลเริ่มต้น เพื่อคำนวณค่าความน่าจะเป็นที่แท้จริงนั้นหนึ่งรอบ ส่วนอีกรอบหนึ่งคือการอ่านไฟล์เพื่อทำการเข้ารหัส โดยเราจะเรียกขั้นตอนเหล่านี้ว่า การอ่านซ้ำสองรอบ (double scan) ดังนั้นการเข้ารหัสที่ค่าความน่าจะเป็นสามารถปรับเปลี่ยนค่าได้ในระหว่างการเข้ารหัสไปพร้อมๆ กัน จึงเป็นทางเลือกอีกทางหนึ่งที่ได้ผลลัพธ์ที่ดีและรวดเร็ว แบบจำลองของการคิดความน่าจะเป็นของสัญลักษณ์แบบปรับเปลี่ยนค่าได้นี้ได้จากอัลกอริทึมที่ 1.3 การปรับเปลี่ยนค่าความน่าจะเป็นของสัญลักษณ์ในระหว่างกระบวนการเข้ารหัสและถอดรหัส

```
Algorithm 1.3: Probability in adaptive model
1   Input:          frequencies of all symbols
2   Output:         probability of all symbols
3   Begin
4       For each Frequency(a) assign frequency of 1
5           N = number of all symbol
6           Do
7               frequency(a) = frequency(a)+1
8               N = N +1
9               probability(a) = frequency(a)/N
10          While(!EOF)
11      End For
12  Return probability
13  End
```

โดยปกติแล้วการเข้ารหัสเลขคณิตด้วยวิธีนี้ จะกำหนดให้ค่าความน่าจะเป็นหรือค่าความถี่เริ่มต้นเริ่มต้นมีค่าเท่ากันในทุกสัญลักษณ์ จากนั้นจึงปรับค่าความน่าจะเป็นไปตามสัญลักษณ์ที่ผ่านมายังตัวเข้ารหัสหรือตัวถอดรหัสด้วยอัลกอริทึม 1.3

นิยาม 2.8 การเข้ารหัสเลขคณิตที่สามารถปรับเปลี่ยนค่าความน่าจะเป็นได้ และกำหนดให้ค่าความน่าจะเป็นเริ่มต้นเท่ากัน (adaptive arithmetic coding with equal initial probability: AAC1) *หมายถึง การเข้ารหัสเลขคณิตที่กำหนดให้มีค่าความน่าจะเป็นของทุกสัญลักษณ์มีค่าเริ่มต้นเท่ากัน ทั้งในตัวเข้ารหัส และ ตัวถอดรหัส นอกจากนั้นยังใช้หลักการของการปรับเปลี่ยนค่าความน่าจะเป็นตามอัลกอริทึมที่ 1.3*

2.1.5 การเข้ารหัสเลขคณิตส่วนเพิ่มขึ้น

ในปัจจุบันการเข้ารหัสโดยการคำนวณเพื่อหาช่วงของจำนวนจริงนั้น สามารถหลีกเลี่ยงปัญหาความแม่นยำในระดับสูง (high precision problem) ในระหว่างการเขียนโปรแกรม จึงนิยมที่จะใช้การเข้ารหัสเลขคณิตส่วนเพิ่มขึ้นแทน

การเข้ารหัสเลขคณิตส่วนเพิ่มขึ้น [4, 9] มีหลักการคือ เมื่อรู้ว่ามีบิตของคำตอบร่วมกัน (common bits) ของช่วงต่ำสุด (low) และ ช่วงสูงสุด (high) ให้ผลิตบิตคำตอบนั้นทันที หากไม่แน่ใจ จะเก็บตัวทดไว้เพื่อส่งบิตคำตอบในรอบถัดไป จากนั้นให้ขยายขนาดของช่วงที่กำลังพิจารณาออกเป็นสองเท่า ซึ่งอัลกอริทึมของการเข้ารหัสเลขคณิตส่วนเพิ่มขึ้นนี้ ได้แสดงไว้ตามอัลกอริทึมที่ 1.4 การเข้ารหัสเลขคณิตส่วนเพิ่มขึ้น และอัลกอริทึมที่ 1.5 การถอดรหัสเลขคณิตส่วนเพิ่มขึ้นตามลำดับ เพื่อให้เข้าใจมากยิ่งขึ้น ได้จึงแสดงตัวอย่างการเข้ารหัสและถอดรหัสเลขคณิตส่วนเพิ่มขึ้นหลังจากอัลกอริทึมที่ 1.4 นี้ด้วย ดังตัวอย่างที่ 2.3

Algorithm 1.4: incremental arithmetic coding

1 **Input:** F ($a_1a_2..a_n$),
array of cumulative frequency of a_i (cum_freq[a_i])
2 **Output:** sequence of X for X ∈ {0,1}
3 **Begin**
4 **Let:** low = 0, high = 1, X = Φ, carry = 0
5 **Do**
6 range = high – low
7 high = low + range * cum_freq[a_i]
8 low = low + range * cum_freq[a_{i-1}]
9 //Consider low, high as subinterval
10 **Do**
11 1.1 If new subinterval is not within [0, ½), [¼ , ¾) or [½, 1), stop and return.
12 1.2 If new subinterval lies in [0, ½), append X with 0 and 1's as the number of carry from step 1.4, low = 2*low and high= 2*high, carry = 0.

13 1.3 If new subinterval lies in [½, 1), append X with 1 and 0's as the number of carry from step 1.4, low = 2*low - 1, high = 2*high – 1, carry = 0.
14 1.4 If new subinterval lies in [¼ , ¾), carry = carry+1, low = 2*low - 0.5, high = 2*high - 0.5.
15 **While** (low and high do not meet any conditions in 1.1 – 1.4)
16 **While** (!EOF)
17 **Return** X
18 **End**

Algorithm 1.5: Incremental arithmetic decoding
1 **Input:** Array of cumulative frequency of a_i (cum_freq[a_i]), Sequence of X for X ∈ {0,1}
2 **Output:** F ($a_1a_2..a_n$)
3 **Begin**
4 **Let:** low = 0, high = 1
5 **Do**
6 Calculate value V in decimal from X
7 V' = (V – low) / (high – low)
8 Find symbol that meet this constraint: cum_freq[a_{i-1}] ≤ V' < cum_freq[a_i]
9 range = high – low
high = low + range * cum_freq[a_i]
10 low = low + range * cum_freq[a_i-1]
//Consider low, high as subinterval
Do
11 1.1 If new subinterval is not within [0, ½), [¼ , ¾) or [½, 1), do nothing.
12 1.2 If new subinterval lies in [0, ½), append F with 0 and 1's as the number of carry from step 1.4,
low = 2*low and high= 2*high, carry = 0, shift X left one bit plus the number of bit from 1's and recalculate value of V from X.
13 1.3 If new subinterval lies in [½, 1), append F with 1 and 0's as the number of carry from step 1.4,
low = 2*low - 1, high = 2*high – 1, carry = 0, shift X left one bit plus the number of bit from 0's and recalculate value of V from X.
14 1.4 If new subinterval lies in [¼ , ¾), carry = carry+1, low = 2*low - 0.5, high = 2*high - 0.5, V = 2*V – 0.5.

```
15              While (low and high do not meet any conditions in 1.1
                – 1.4)
16          While (!EOF)
17    Return X
18    End
```

ตัวอย่างที่ 2.3 การเข้ารหัสและถอดรหัสเลขคณิตส่วนเพิ่มขึ้น

กำหนดให้ค่าความน่าจะเป็นของสัญลักษณ์ {a, b, c} แสดงดังในตารางที่ 2.2 ต้องการเข้ารหัสของไฟล์ข้อมูลที่ประกอบไปด้วยกลุ่มของสัญลักษณ์ bbbc ขั้นตอนในการเข้ารหัสเลขคณิตส่วนเพิ่มขึ้นซึ่งได้แสดงดังในตารางที่ 2.5 ขั้นตอนในการถอดรหัสเลขคณิตส่วนเพิ่มขึ้นซึ่งได้แสดงดังในตารางที่ 2.6 ตามลำดับ

ตารางที่ 2.5 การเข้ารหัสเลขคณิตส่วนเพิ่มขึ้น

ช่วงปัจจุบัน	งาน	ช่วงย่อย			สัญลักษณ์
		a	b	c	
[0.0, 1.0)	แบ่ง	[0.0, 0.4)	[0.4, 0.9)	[0.9, 1)	b
[0.4, 0.9)	แบ่ง	[0.4, 0.6)	[0.6, 0.85)	[0.85, 0.9)	b
[0.6, 0.85)	ออกบิต 1 ขยายช่วง				
[0.20, 0.70)	แบ่ง	[0.20, 0.40)	[0.40, 0.65)	[0.65, 0.70)	b
[0.40, 0.65)	ทด ขยายช่วง				
[0.3, 0.8)	แบ่ง	[0.30, 0.50)	[0.50, 0.75)	[0.75, 0.80)	c
[0.75, 0.80)	ออกบิต 10 ขยายช่วง				
[0.50, 0.60)	ออกบิต 1 ขยายช่วง				
[0.00, 0.20)	ออกบิต 0 ขยายช่วง				
[0.00, 0.40)	ออกบิต 0 ขยายช่วง				
[0.00, 0.80)	ออกบิต 0 ขยายช่วง หยุด				

จากตัวอย่างที่ 2.2 ในตารางที่ 2.3 พบว่าค่าช่วงของคำตอบของการเข้ารหัสคือ [0.8125, 0.825) เมื่อเขียนในรูปเลขฐานสองจะได้ค่าประมาณ [0.11010 00000, 0.11010 01100) จากตารางที่ 2.5 พบว่าค่าความยาวที่น้อยที่สุดของบิตคำตอบในการเข้ารหัสนี้สามารถคำนวณได้จากสมการ (3) โดยค่าของ $\sum_{i=1}^{n} lg \frac{1}{P(x_i)}$ เมื่อแทนด้วยค่าความน่าจะเป็นของ b, b, b และ c ตามลำดับ จะได้ค่าเท่ากับ $-\lg(0.5) - \lg(0.5) - \lg(0.5) - \lg(0.1) = \lg(0.5)^3(0.1) = 6.321928 \approx$ 7 บิต ดังนั้นเราจึงเลือกคำตอบที่ให้จำนวนบิตที่สั้นที่สุดคือ 0.1101000 (0.8125) ซึ่งตรงกับตัวอย่างที่ 2.2

ตารางที่ 2.6 การถอดรหัสเลขคณิตส่วนเพิ่มขึ้น

บิตคำตอบ	ช่วงปัจจุบัน	V	V'	สัญลักษณ์	งาน
0.1101000	[0,1)	0.8125	(0.8125-0)/(1-0) = 0.8125	b	แบ่ง
	L: 0+(1-0)0.4 = 0.4 H: 0+(1-0)0.9 = 0.9		(0.8125-0.4)/(0.9-0.4) = 0.825	b	แบ่ง
0.1101000	L: 0.4+(0.9-0.4)0.4 = 0.6 H: 0.4+(0.9-0.4)0.9 =0.85	0.625			ออกบิต 1 ขยายช่วง
	L: 0.6(2)-1 = 0.2 H: 0.85(2)-1 = 0.7		(0.625-0.2)/(0.7-0.2) =0.85	b	แบ่ง
	L: 0.2+(0.7-0.2)0.4 = 0.4 H: 0.2+(0.7-0.2)0.9 = 0.65				ทด ขยายช่วง
	L: 2(0.4)-0.5 = 0.3 H: 2(0.65)-0.5 = 0.8	2*0.625-0.5 = 0.75	(0.75-0.3)/(0.8-0.3) = 0.9	c	แบ่ง
0.11001000	L: 0.3+(0.8-0.3)0.9 = 0.75 H: 0.3+(0.8-0.3)1 = 0.8				ออกบิต 10 ขยายช่วง
0.11011000	L: 0.75*2-1 = 0.5 H: 0.8*2-1 = 0.6				ออกบิต 1 ขยายช่วง
0.11010000	L: 0.5*2-1 = 0 H: 0.6*2-1 = 0.2				ออกบิต 0 ขยายช่วง
0.11010000	L: 2*0 = 0 H: 0.2*2 = 0.4				ออกบิต 0 ขยายช่วง
0.1101000	L: 2*0 = 0 H: 0.4*2 = 0.8				ออกบิต 0 หยุด

การถอดรหัสเลขคณิตส่วนเพิ่มขึ้นนี้ใช้หลักการที่เหมือนกับการถอดรหัสเลขคณิตตรงที่ การแบ่งส่วนย่อยและการนำค่า V' ไปค้นหาว่ามีค่าความน่าจะเป็นอยู่ในช่วง low (L) และ high (H) ใด นอกจากนั้นเมื่อมีการออกบิต 0 หรือ 1 ไป ตามสดมภ์ชื่อ 'งาน' แล้ว ให้เลื่อนบิตคำตอบไปทางซ้าย (shift left) หรือการขยายช่วงนั่นเอง ในที่นี้ใช้การขีดเส้นใต้เพิ่มขึ้นตามจำนวนบิตที่ออกค่าไปแทนการขยายช่วง ดังนั้นค่าของ V ณ รอบของการถอดรหัสใดๆ คิดค่า ได้จากบิตที่เหลืออยู่หรือบิตที่ไม่ขีดเส้นใต้ ตัวอย่างเช่น 0.1101000 จะมีค่าเท่ากับ 0.101000 หรือเท่ากับ 1/2 + 1/8 = 0.625 นอกจากนี้หากรอบการคำนวณก่อนหน้ามีการทดไว้ ค่า V ในรอบปัจจุบันจะคิดได้จาก 2*V - 0.5 แทน

□

2.1.6 การประมาณค่าความน่าจะเป็นสำหรับการเข้ารหัสเลขคณิต

จากตัวอย่างที่ 2.1 พบว่าการประมาณค่าความน่าจะเป็นที่ดี จะส่งผลทำให้ค่าเอนโทรปีของต้นแบบนั้นมีค่าลดลง ซึ่งจะนำไปสู่การได้ค่าอัตราส่วนการบีบอัดที่ดีขึ้นตามมาด้วย พิจารณาตัวอย่างดังต่อไปนี้

ตัวอย่างที่ 2.4 กำหนดให้ $X = abaabcda$ บนเซตอักขระ $\{a, b, c, d\}$ สมมติให้มีผลลัพธ์ในการประมาณค่าความน่าจะเป็นของแต่ละต้นแบบซึ่งมีความสัมพันธ์ตาม

$$P_{M_1}(x) = P(x)\ \forall x \in A \coloneqq \{a, b, c, d\}$$

ไว้ดังต่อไปนี้

ต้นแบบที่หนึ่ง:

$$P_{m_1}(a) = 0.5, P_{m_1}(b) = 0.25, P_{m_1}(c) = 0.125, P_{m_1}(c) = 0.125$$

ต้นแบบที่สอง:

$$P_{m_1}(a) = 0.125, P_{m_1}(b) = 0.125, P_{m_1}(c) = 0.5, P_{m_1}(c) = 0.25$$

ค่าเอนโทรปีของต้นแบบที่หนึ่ง และ ต้นแบบที่สองสามารถคำนวณได้ดังต่อไปนี้

$$\begin{aligned}
H_{M_1} &= \sum_{x\in\{a,b,c,d\}} P(x)\, lg \frac{1}{P_{M_1}(x)} \\
&= (0.5 * lg\frac{1}{0.5}) + (0.25 * lg\frac{1}{0.25}) + (0.125 * lg\frac{1}{0.125}) \\
&\quad + (0.125 * lg\frac{1}{0.125}) \\
&= 0.5lg2 + 0.25lg4 + 0.125lg8 + 0.125lg8 \\
&= 1.75\frac{bit}{symbol}
\end{aligned}$$

$$\begin{aligned}
H_{M_2} &= \sum_{x\in\{a,b,c,d\}} P(x)\, lg \frac{1}{P_{M_2}(x)} \\
&= (0.5 * lg\frac{1}{0.125}) + (0.25 * lg\frac{1}{0.25}) + (0.125 * lg\frac{1}{0.5}) \\
&\quad + (0.125 * lg\frac{1}{0.25}) \\
&= 0.5lg8 + 0.25lg8 + 0.125lg2 + 0.125lg4 \\
&= 2.625\frac{bit}{symbol}
\end{aligned}$$

□

จากค่าเอนโทรปีของต้นแบบที่หนึ่งและสองได้ค่าเท่ากับ 1.75 และ 2.625 bit/symbol ตามลำดับนั้น จะใช้จำนวนบิตทั้งหมดเท่ากับ 1.75*8 = 14 บิต และ 2.625*8 = 21 บิต ซึ่งต้นแบบที่สองนั้นใช้บิตมากกว่า จึงทำให้ค่าอัตราส่วนการบีบอัดมีค่าน้อยกว่าเมื่อเทียบกับแบบที่หนึ่ง

นอกจากนั้นจำนวนบิตที่ดีที่สุดโดยรวมนั้น หากใช้สมการ (3) จะมีค่าเท่ากับ $4 * \lg\left(\frac{4}{8}\right) + 2 * \lg\left(\frac{2}{8}\right) + 1 * \lg\left(\frac{1}{8}\right) + 1 * \lg\left(\frac{1}{8}\right) = 14$ บิตซึ่งตรงกันกับเมื่อใช้ค่าเอนโทรปีในการคิดคำนวณ นอกจากนั้นค่าความน่าจะเป็นที่ได้จากต้นแบบที่หนึ่งนั้นพบว่าเป็นค่าความน่าจะเป็นที่แท้จริง (exact probability of symbol) ที่ได้มาจากการอ่านซ้ำสองรอบของแต่ละสัญลักษณ์ในสายอักขระ $abaabcda$ นั่นเอง

การอ่านซ้ำสองรอบนี้เอง จึงเป็นที่มาของการประมาณค่าความน่าจะเป็นของสัญลักษณ์เพื่อลดความซ้ำซ้อนในการอ่านไฟล์และเพื่อให้จำนวนบิตของคำตอบมีค่าใกล้เคียงกับค่าเอนโทรปีที่ดีที่สุดที่ได้จากการอ่านค่าความน่าจะเป็นที่แท้จริง จึงต้องอาศัยการประมาณค่าความจะเป็นของสัญลักษณ์ที่เหมาะสมอีกด้วย วิธีในการประมาณค่าความน่าจะเป็นของสัญลักษณ์นั้นมีอยู่ด้วยกันหลายวิธี ซึ่งจะขอกล่าวถึงเนื้อหาในส่วนนี้ในหัวข้อถัดๆไป

2.1.7 ค่าอัตราส่วนการบีบอัด

ค่าอัตราส่วนการบีบอัดคือ อัตราส่วนระหว่างจำนวนบิตนำเข้า ($|bitin|$) หรือขนาดของข้อมูลนำเข้า ต่อจำนวนของบิตคำตอบ ($|bitout|$) ต่อ แต่ในงานวิจัยคุโรกิและคณะ (Kuroki et. al.) [2] ได้รวมเอาขนาดของพจนานุกรม ($|table|$) เข้าไปในการคิดค่าอัตราส่วนการบีบอัดด้วย ซึ่งขนาดของพจนานุกรมจะมีค่าเป็นศูนย์ ในกรณีของการเข้ารหัสและถอดรหัสเลขคณิต นอกจากนี้ค่าอัตราส่วนการบีบอัดจะมีขนาดหนึ่งเมื่อใช้การเข้ารหัสโดยอาศัยพจนานุกรม ดังนั้นสูตรในการคำนวณค่าอัตราส่วนการบีบอัดจึงเป็นดังสมการ (4)

$$CR = \frac{Original\ size}{Compressed\ size} = \frac{|bitin|}{|bitout|}, \frac{|bitin|}{|bitout|+|table|} \tag{4}$$

2.1.8 การจัดกลุ่ม (clustering)

จากงานวิจัยของเจนและคณะ (Jain et. al.) [31] ได้ให้ความหมายของการจัดกลุ่มไว้ว่า การจัดกลุ่มเป็นการจำแนกในข้อมูลที่ไม่ได้มีการระบุฉลาก (unlabeled) และยังไม่ได้ตรวจตรา (unsupervised) ให้ออกเป็นกลุ่มๆ (cluster) ที่มีความคล้ายคลึงกัน แต่ถ้าเราทราบถึงประเภทที่แน่ชัดของข้อมูลแล้ว เราจะเรียกวิธีการนั้นว่าการจำแนกประเภท (classification)

2.1.8.1 เทคนิคในการจัดกลุ่ม (clustering techniques)

เทคนิคในการจัดกลุ่มของข้อมูลสามารถแบ่งออกได้ตามหลักการในการทำงาน ได้เป็นสองชนิดคือ การใช้ความสัมพันธ์ตามลำดับชั้น (hierarchical clustering) และการแบ่งแยก (partitioned clustering)

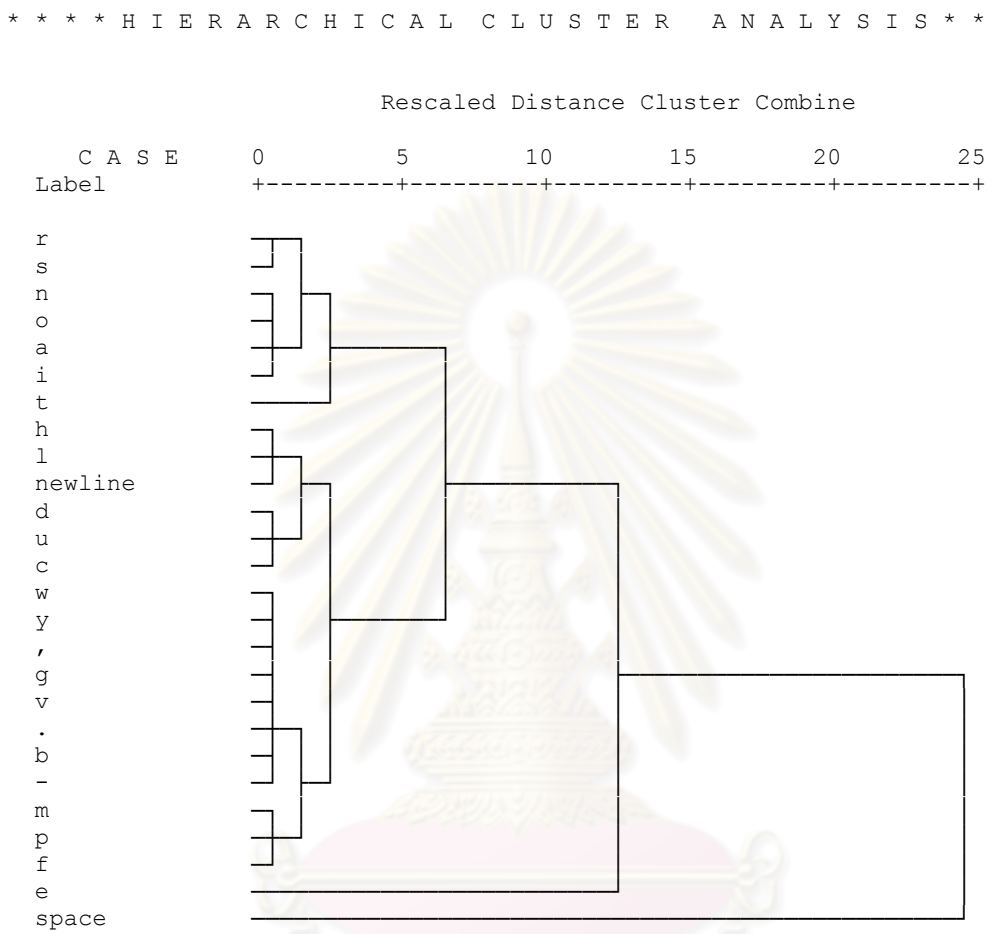
2.1.8.1.1 การใช้ความสัมพันธ์ตามลำดับชั้น (hierarchical clustering)

การใช้ความสัมพันธ์ตามลำดับชั้นเป็นเทคนิคที่นิยมกันมากในการจัดกลุ่มโดยการพิจารณาจากแนวโน้มของความสามารถในการอยู่รวมกัน (agglomerative) ของข้อมูลที่สนใจ โดยเริ่มต้นจากการแบ่งกลุ่มทั้งหมดให้เท่ากับจำนวนของข้อมูลที่สนใจทั้งหมด จากนั้นลดทอนจำนวนกลุ่มลงด้วยการรวมกลุ่มอย่างมีลำดับและขั้นตอน (successively) โดยใช้การพิจารณาค่าระยะห่าง (distance metrics) ของข้อมูลทีละคู่ว่ามีระดับความเหมือนกัน (similarity level) ว่าอยู่ในกลุ่มเดียวกันหรือไม่ โดยในแต่ละขั้นตอนอาจมีการรวมข้อมูลที่สนใจไปกับกลุ่มที่มีอยู่แล้ว หรือรวมกลุ่มสองกลุ่มเข้าไปไว้ในกลุ่มใหม่ จากนั้นให้รวมกลุ่มจนกระทั่งมีกลุ่มที่ใหญ่ที่สุดเพียงกลุ่มเดียว ซึ่งคือกลุ่มที่ประกอบด้วยข้อมูลที่สนใจทุกตัว การใช้ความสัมพันธ์ตามลำดับชั้นนิยมใช้เดนโดรแกรม (dendrogram) เพื่อแสดงถึงความสัมพันธ์ในการรวมกลุ่มกันของข้อมูลใดๆ ที่มีระดับความเหมือนอยู่ในระดับเดียวกัน ซึ่งได้แสดงไว้ดังรูปที่ 2.2 เดนโดรแกรมของความถี่ของสัญลักษณ์ในเอกสารภาษาอังกฤษบางสัญลักษณ์

นิยาม 2.9 เอกสารข้อความภาษาอังกฤษ หรือ เอกสารภาษาอังกฤษ *หมายถึงเอกสารที่มีรหัสแอสกีที่เมื่อพิมพ์โดยใช้คีย์บอร์ดในภาษาอังกฤษแล้ว สามารถปรากฏได้ในเอกสารนั้น เช่น สัญลักษณ์ a – z และ A - Z เครื่องหมายวรรคตอนต่างๆ (puncation) ตัวเลข สัญลักษณ์ขึ้นบันทัดใหม่ เป็นต้น*

รูปที่ 2.2 เป็นเดนโดรแกรมของความถี่ในสัญลักษณ์แอสกีบางตัว ในเอกสารภาษาอังกฤษพบว่าที่ระดับระยะห่างระหว่างกลุ่มที่แปลงค่าแล้ว (rescale distance cluster combine) ที่ระดับต่ำสุด ที่ระดับ 0 นั้น สัญลักษณ์ที่อยู่ในกลุ่มเดียวกัน คือสัญลักษณ์ที่มีค่าความถี่ใกล้เคียงกันมาก โดยอาศัยการพิจารณาจากเส้นที่เชื่อมถึงกัน โดยจากรูป สัญลักษณ์ ‘r’ และ ‘s’ จัดอยู่ในกลุ่มเดียวกัน สัญลักษณ์ ‘e’ จะอยู่คนละกลุ่มกับสัญลักษณ์ ‘space’ นอกจากนั้นที่ระดับระยะห่างระหว่างกลุ่มที่แปลงค่าแล้วที่มีค่ามากขึ้น ค่าความถี่ของแต่ละสัญลักษณ์จะมีค่าความถี่ที่มีความแตกต่างกันมากขึ้น และจากการพิจารณาเส้นเชื่อมถึงกันแล้ว กลุ่มของสัญลักษณ์ย่อยๆ จะรวมกลุ่มกันเป็นอีกกลุ่มใหญ่ และ

จะรวมกันไปในลักษณะเช่นนี้เรื่อยๆ จนกระทั่งสุดท้ายมีกลุ่มใหญ่แค่เพียงกลุ่มเดียว หรือเป็นกลุ่มที่ประกอบด้วยสัญลักษณ์ทั้งหมดในรหัสแอสกี จำนวน 256 สัญลักษณ์



รูปที่ 2.2 เดนโดรแกรมของความถี่ของสัญลักษณ์ในเอกสารภาษาอังกฤษบางสัญลักษณ์

2.1.8.1.2 การแบ่งแยก (partitioned clustering)

ในปัจจุบัน การจัดกลุ่มโดยใช้แบ่งแยกที่เป็นที่นิยมมากที่สุดวิธีหนึ่ง คือ การแบ่งแยกโดยใช้ค่าเฉลี่ยเค (k-means clustering) [32 - 34] โดยที่ค่าเคเป็นจำนวนกลุ่มที่ต้องการ วิธีการนี้จะแตกต่างจากการจัดกลุ่มโดยการใช้ความสัมพันธ์ตามลำดับชั้น โดยเริ่มจากจำนวนกลุ่มที่ระบุไว้อย่างชัดเจน และคำนวนหาว่าข้อมูลแต่ละข้อมูลอยู่ใกล้กับจุดกึ่งกลาง (centroid) ของกลุ่มใดมากที่สุด ให้จัดข้อมูลนั้นอยู่ในกลุ่มนั้น

หลักการของการแบ่งแยกโดยการใช้ค่าเฉลี่ยเคนี้สามารถอธิบายเป็นขั้นตอนๆ ที่สำคัญคือ

- เลือกจำนวนกลุ่มที่ต้องการ
- กำหนดจุดแทนตำแหน่งของแต่ละกลุ่มโดยวิธีการสุ่ม (cluster centroid)
- จัดกลุ่มของข้อมูลทุกตัว ที่อยู่ใกล้กับจุดแทนตำแหน่งใดๆ มากที่สุด
- คำนวณจุดแทนตำแหน่งจากข้อมูลทุกตัวที่อยู่ในกลุ่มเดียวกัน โดยมีสมมติฐานที่ว่าสมาชิกในกลุ่มแต่ละกลุ่มนั้นถูกต้อง
- ถ้าไม่มีการเปลี่ยนแปลงการเป็นสมาชิกของกลุ่มเกิดขึ้นในการคำนวณรอบสุดท้ายให้หยุดการคำนวณ และคืนค่าสมาชิกในกลุ่มแต่ละกลุ่ม มิฉะนั้นให้กลับไปทำขั้นตอนที่ 3 ใหม่อีกครั้ง

2.1.9 การพิจารณาเลือกจำนวนกลุ่มที่เหมาะสม

เราสามารถพิจารณาเลือกจำนวนกลุ่มที่เหมาะสมได้ด้วยการนับจำนวนกลุ่มในเดนโดรแกรมที่มีระดับระยะห่างระหว่างกลุ่มที่แปลงค่าแล้วที่ระดับต่ำสุดที่ระดับ 0

2.1.10 การกระจายแบบไวบูลล์ (Weibull distribution)

การกระจายแบบไวบูลล์ ได้ถูกตั้งชื่อตาม วาโลดี ไวบูลล์ (Waloddi Weibull) ผู้ที่ค้นคิดการกระจายชนิดนี้ในปี ค.ศ. 1951 การกระจายแบบนี้มีความยืดหยุ่นสูง และมักจะเป็นที่อ้างถึงสำหรับการศึกษาทางด้านวิศวกรรมและแบบจำลองสถิติ โดยเฉพาะอย่างยิ่งได้ถูกนำไปใช้อ้างอิงในการวิเคราะห์ความน่าเชื่อถือ (reliability analysis) [35] และอายุขัยของเครื่องจักร สามารถกำหนดค่าอัตราความล้มเหลว (failure rate) ที่แตกต่างกันรูปของฟังก์ชันเวลา โดยลักษณะของอัตราความล้มเหลวแบ่งออกได้เป็น 3 ลักษณะดังต่อไปนี้

- กรณีที่ 1 หมายถึงว่า ค่าอัตราความล้มเหลวจะมีค่าลดลง เมื่อเวลาเปลี่ยนไป
- กรณีที่ 2 หมายถึงว่า ค่าอัตราความล้มเหลวจะมีค่าคงที่ เมื่อเวลาเปลี่ยนไป
- กรณีที่ 3 หมายถึงว่า ค่าอัตราความล้มเหลวจะมีค่าเพิ่มขึ้น เมื่อเวลาเปลี่ยนไป

ผู้วิจัยขอนำเสนอนิยามเบื้องต้นประกอบการทำความเข้าใจสำหรับการกระจายแบบไวบูลล์ ไว้ดังต่อไปนี้

นิยาม 2.10 *ตัวแปรสุ่ม (random variable) คือ ฟังก์ชันจับคู่ (mapping function) ระหว่างเหตุการณ์ที่เป็นไปได้ในปริภูมิตัวอย่าง (sample space: S) และค่าที่เป็นจำนวนจริง โดยทั่วไป เราใช้สัญลักษณ์ภาษาอังกฤษตัวใหญ่ X, Y, Z เป็นสัญลักษณ์แทนตัวแปรสุ่ม ส่วนค่าของตัวแปรสุ่มดังกล่าว แทนด้วยสัญลักษณ์ภาษาอังกฤษตัวเล็ก คือ x, y, z ตามลำดับ นั่นคือ $X: S \to R$ หรือ $X(s) = x$ โดยที่ $x \in R$*

นิยาม 2.11 *ตัวแปรสุ่มที่เป็นอิสระต่อกันและกระจายตัวอย่างเท่าเทียมกัน (independent identically distributed random variable) จะเป็นตัวแปรสุ่มที่ทุกๆ ตัวแปรสุ่มจะมีการกระจายของความน่าจะเป็นในรูปแบบเดียวกัน และความน่าจะเป็นที่จะปรากฏตัวแปรสุ่มแต่ละตัวนั้นเป็นอิสระต่อกัน*

ประเภทของตัวแปรสุ่ม

- ตัวแปรสุ่มแบบไม่ต่อเนื่อง (discrete random variable) หมายถึงตัวแปรสุ่ม ที่มีค่าของตัวแปรสุ่มแบบไม่ต่อเนื่องกัน ส่วนมากเป็นค่าที่เป็นเลขจำนวนนับ ได้แก่การโยนเหรียญ การทอยลูกเต๋า เป็นต้น
- ตัวแปรสุ่มแบบต่อเนื่อง (continuous random variable) หมายถึงตัวแปรสุ่ม ที่มีค่าของตัวแปรสุ่ม ที่เกี่ยวข้องกับมาตราชั่งตวงวัด ได้แก่ น้ำหนัก ส่วนสูง ระยะเวลา อุณหภูมิ เป็นต้น โดยจะมีค่าต่อเนื่องกัน และมีจำนวนไม่จำกัด

นิยาม 2.12 *ฟังก์ชันความหนาแน่นของความน่าจะเป็น (probability density function: PDF) หรือฟังก์ชันความน่าจะเป็น (probability function) ของตัวแปรสุ่มแบบไม่ต่อเนื่อง X นั้น ถูกกำหนดโดย ฟังก์ชัน f โดยที่*

$$f(x_i) = P(X = x_i) = P(x_i)$$

เมื่อ $i = 1, 2, 3..$
และ

$$P(x_i) = \frac{n(x_i)}{n(S)}$$

นอกจากนั้น $f(x_i)$ ยังมีคุณสมบัติดังต่อไปนี้

- $f(x_i) \geq 0;\ \forall i$
- $\sum_i f(x_i) = 1$

นิยาม 2.13 *ฟังก์ชันความหนาแน่นของความน่าจะเป็น (probability density function: PDF) หรือฟังก์ชันความน่าจะเป็น (probability function) ของตัวแปรสุ่มแบบต่อเนื่อง X นั้น ถูกกำหนดโดย ฟังก์ชัน f โดยมีคุณสมบัติดังต่อไปนี้*

- $f(x) \geq 0$
- $\int_{-\infty}^{\infty} f(x)dx = 1$

นิยาม 2.14 *ตัวแปรสุ่มใดๆ จะมีการกระจายแบบไวบูลล์ ซึ่งประกอบด้วยพารามิเตอร์แสดงรูปร่าง (shape parameter - α) และพารามิเตอร์มาตราส่วน (scale parameter - β) โดยที่ ($\alpha > 0$, $\beta > 0$) ถ้า PDF ของตัวแปรสุ่มแบบต่อเนื่อง X นั้นมีสมบัติเป็นไปตามสมการ*

$$f(x,\alpha,\beta) = \begin{cases} \frac{\alpha}{\beta}\left(\frac{x}{\beta}\right)^{\alpha-1} e^{-\left(\frac{x}{\beta}\right)^{\alpha}} & x \geq 0 \\ 0 & x < 0. \end{cases}$$

ในกรณีของการกระจายแบบไวบูลล์ เราสามารถคำนวณหาฟังก์ชันผกผัน (Inverse Function) ได้ตามสมการ

$$x = F(y) = \int_0^y f(z,\alpha,\beta)dz.$$

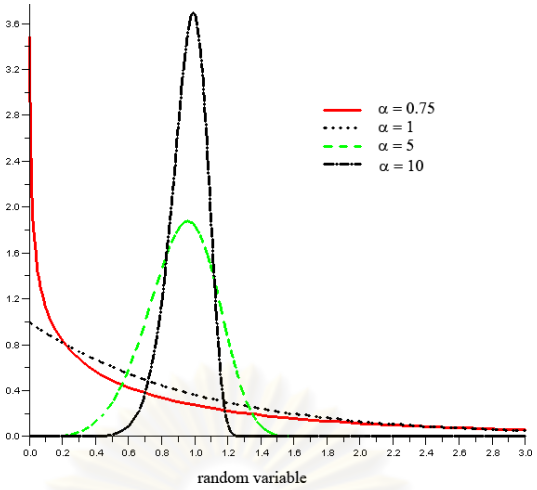
จากนั้นเราแก้สมการอินทริเกรชันดังกล่าว เราจะได้

$$x = F(y) = 1 - e^{-\left(\frac{y}{\beta}\right)^{\alpha}}.$$

จากนั้นเราจะได้ฟังก์ชันก่อกำเนิด (generating function) ของการกระจายแบบไวบูลล์ดังดังสมการ

$$F = G(x) = -\beta[\ln(1-x)]^{\frac{1}{\alpha}} \quad (5)$$

รูปที่ 2.3 แสดงถึงลักษณะรูปร่างต่างๆ กันของการกระจายแบบไวบูลล์ โดยที่ค่าพารามิเตอร์มาตราส่วน มีค่าเป็น 1 ค่าพารามิเตอร์แสดงรูปร่าง มีค่าเป็น 0.75, 1, 5 และ 10 ตามลำดับ



รูปที่ 2.3 ฟังก์ชันความหนาแน่นของความน่าจะเป็นของการกระจายแบบไวบูลล์ (β = 1)

2.1.11 จำนวนของอันตรภาคชั้นของฮีสโตแกรม (bin size)

สเติร์จจิส (Sturges H. A.) [36] ได้นำเสนอสูตรในการคำนวณหาจำนวนอันตรภาคชั้น หรือ bin สำหรับฮีสโตแกรมไว้ แต่อย่างไรก็ดีในแบบจำลองของเขานั้นสามารถนำไปใช้ได้กับข้อมูลที่มีการกระจายเป็นแบบปกติ (normal distribution) ที่ไม่สามารถปรับเปลี่ยนค่าความเบ้ (skewness) ได้เท่านั้น ต่อมา โดน (Doane D. P.) [37] จึงได้พยามที่จะปรับแบบจำลองของสเติร์จจิส ให้สามารถมีความยืดหยุ่น เพื่อรองรับการคำนวณหาจำนวนอันตรภาคชั้นสำหรับการกระจายรูปแบบต่าง ๆ ได้ นอกจากนั้น แวน (Wand M. P.) [38] ยังได้เสนอว่า แทนที่จะมุ่งเน้นไปทางด้านการคำนวณของจำนวนอันตรภาคชั้น ให้คำนวณหาความกว้างของอันตรภาคชั้น (bin width) ที่เหมาะสมนั้นแทน แต่อย่างไรก็ตามการศึกษาเพื่อหาจำนวนอันตรภาคชั้นที่เหมาะสม สำหรับการกระจายแบบไวบูลล์ ที่เอื้อประโยชน์ต่อการเข้ารหัสเลขคณิตนั้นอยู่นอกเหนือไปจากขอบเขตการวิจัยในครั้งนี้ จึงทำให้ผู้วิจัย นำเอาสมการในการหาจำนวนของอันตราภาคชั้นของฮีสโตแกรม ของ สเติร์จจิสมาใช้ ซึ่งสมการในการคำนวณนี้ เป็นไปดังสมการ (6) โดยที่ k เป็นจำนวนของอันตรภาคชั้น และ n เป็นจำนวนของข้อมูลที่แตกต่างกัน

$$k = 1 + \log_2 n \tag{6}$$

2.1.13 การประมาณค่าความน่าจะเป็นของสัญลักษณ์

วิธีในการประมาณค่าความน่าจะเป็นของสัญลักษณ์สำหรับการเข้ารหัสนั้นมีได้หลากหลายวิธี ซึ่งสองวิธีที่เป็นที่นิยมในกลุ่มนักวิจัยงานทางด้านนี้ คือการ

ประมาณค่าความน่าจะเป็นด้วยทฤษฎีของเบส์ และ ฟังก์ชันความหนาแน่นของความน่าจะเป็น ในที่นี้จะกล่าวถึงการประมาณค่าความน่าจะเป็นด้วยทฤษฎีของเบส์ ส่วนการประมาณค่าความน่าจะเป็นด้วยฟังก์ชันความหนาแน่นของความน่าจะเป็นจะนำเสนอไว้ในบทที่ 3

การประมาณค่าความน่าจะเป็นด้วยทฤษฎีของเบย์

จากทฤษฎีของเบย์ (Bayes' theorem) ได้กล่าวไว้ว่า

$$P(A|B) = \frac{P(B|A) * P(A)}{P(B)} = \frac{P(A \cap B)}{P(A)}$$

โดยที่ $P(A|B)$ คือ ความน่าจะเป็นแบบมีเงื่อนไขของเหตุการณ์ A ที่จะเกิดขึ้นในภายหลัง (posterior probability) เมื่อเกิดเหตุการณ์ B แล้ว

$P(B|A)$ คือ ความน่าจะเป็นแบบมีเงื่อนไขของเหตุการณ์ B ที่จะเกิดขึ้นในภายหลัง เมื่อเกิดเหตุการณ์ A แล้ว

$P(A)$ คือ ความน่าจะเป็นเริ่มต้นของเหตุการณ์ A (prior probability) ที่ทราบโดยไม่ต้องอาศัยข้อมูลเพิ่มเติม

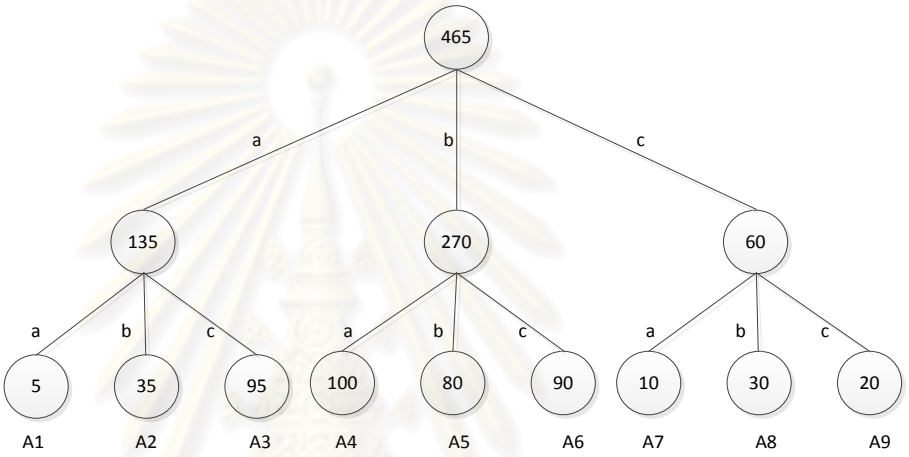
$P(B)$ คือ ความน่าจะเป็นเริ่มต้นของเหตุการณ์ B ที่ทราบโดยไม่ต้องอาศัยข้อมูลเพิ่มเติม

$P(A \cap B)$ คือ ความน่าจะเป็นของการเกิดเหตุการณ์ A ร่วมกับเหตุการณ์ B

ดังนั้นเราสามารถนำมาประยุกต์ในการหาค่าความน่าจะเป็นตามทฤษฎีของเบย์ได้ โดยการศึกษาจากตัวอย่างดังต่อไปนี้

ตัวอย่างที่ 2.5 การคำนวณค่าเอนโทรปี

กำหนดให้ $X = A3\ A1\ A5$ บนเซตอักขระ $\{A1, A2, .., A9\}$ และให้ความถี่ของแต่ละเซตอักขระตามลำดับดังต่อไปนี้ 5, 35, 95, 100, 80, 90, 10, 30 และ 20 จากนั้นสามารถนำไปสร้างต้นไม้ตติยภูมิ (tertiary tree) โดยที่โหนดใบ (leaf node) บรรจุค่าความถี่ ส่วนโหนดที่ไม่ใช่โหนดใบ (non – leaf node) บรรจุค่าความถี่รวมของโหนดใบที่เป็นลูกเหล่านั้น กำหนดให้เซตอักขระย่อย $\{a, b, c\}$ เป็นเซตอักขระที่แทนเส้นทางกำกับจากโหนดแม่ (root node) ไปยังโหนดใบใดๆ ดังแสดงไว้ดังรูปที่ 2.4



รูปที่ 2.4 ต้นไม้ตติยภูมิของเซตอักขระ $\{A1, A2, .., A9\}$

สำหรับการเข้ารหัส $X = A3\ A1\ A5$ เราจะได้เซตอักขระย่อย ac aa bb ตามลำดับ และพิจาณาการคิดค่าความน่าจะเป็นแบบมีเงื่อนไขตามทฤษฎีของเบย์ ดังแสดงไว้ในตารางที่ 2.7

ตารางที่ 2.7 ตัวอย่างการประมาณค่าความน่าจะเป็นโดยทฤษฎีของเบย์

สัญลักษณ์	ลำดับสัญลักษณ์ที่พิจารณา	ความน่าจะเป็นโดยทฤษฎีเบย์
a	$P(a \mid \Lambda)$	$\frac{135}{465}$
c	$P(c \mid a)$	$\frac{P(c \cap a)}{P(a)} = \frac{95/465}{135/465} = \frac{95}{135}$
a	$P(a \mid \Lambda)$	$\frac{135}{465}$
a	$P(a \mid a)$	$\frac{P(a \cap a)}{P(a)} = \frac{5/465}{135/465} = \frac{5}{135}$
b	$P(b \mid \Lambda)$	$\frac{270}{465}$
b	$P(b \mid \mathrm{b})$	$\frac{P(b \cap b)}{P(b)} = \frac{80/465}{270/465} = \frac{80}{270}$

□

2.2 งานวิจัยที่เกี่ยวข้อง

การเปลี่ยนโครงสร้างของข้อมูลให้อยู่ในรูปแบบใดรูปแบบหนึ่ง การเตรียมข้อมูลก่อนการเข้ารหัสและการประมาณค่าความน่าจะเป็นของสัญลักษณ์ ต่างมีจุดประสงค์เพื่อเอื้ออำนวยให้ใช้ทรัพยากรที่น้อยลง ได้อัตราส่วนการบีบอัดที่ดีขึ้น มักจะเป็นที่นิยมเสมอตลอดระยะเวลาที่ผ่านมาในงานวิจัยที่เกี่ยวข้องกับการเข้ารหัสและถอดรหัสเลขคณิต การเปลี่ยนแปลงโครงสร้างของข้อมูลให้อยู่ในรูปแบบใดรูปแบบหนึ่งมีอยู่หลายวิธีด้วยกัน เช่น ในปี ค.ศ. 2007 แอพพาราจูและคณะ (Apparaju et. al.) [1] ได้นำเสนอการแปลงข้อมูลนำเข้าให้เป็นกลุ่มสัญลักษณ์ (multi-symbols) แล้วจัดกลุ่มสัญลักษณ์เหล่านั้นให้อยู่ในโครงสร้างของต้นไม้แบบ m ภาค (m-ary tree) และใช้ทฤษฎีของเบย์ มาใช้ในการคิดค่าความน่าจะเป็นของกลุ่มสัญลักษณ์ตามโครงสร้างของต้นไม้ จากนั้นจึงการเข้ารหัสและถอดรหัสเลขคณิตกับคลังข้อมูลแคลการี งานวิจัยได้รายงานผลว่า ค่า m ที่ให้ค่าอัตราส่วนการบีบอัดที่ดีขึ้นอยู่ระหว่างค่า 2 ถึง 4

ต่อมาในปี ค.ศ. 2008 โคชะเน็คและคณะ (Kochanek et. al.) [20] ใช้การแบ่งข้อมูลนำเข้าออกเป็นส่วนย่อยๆ (multi-stream) เพื่อแปลงข้อมูลในแต่ละส่วนย่อยให้แสดงผลในรูปของต้นไม้ที่ต่างไปจากต้นไม้ของฮัฟฟ์แมน (Huffman tree) โดยในแต่ละโหนดใบ (node) ประกอบไปด้วยสามลักษณะ (attribute) ได้แก่ สัญลักษณ์หรือกลุ่มของสัญลักษณ์ ความถี่รวมของแต่ละสัญลักษณ์ในกลุ่มของสัญลักษณ์ และตำแหน่งที่ปรากฏครั้งแรกของสัญลักษณ์หรือกลุ่มของสัญลักษณ์ จากนั้นใช้โครงสร้างต้นไม้แบบนี้ร่วมกับการเข้ารหัสแบบอื่นๆ จากผลการทดลองระบุว่า การเข้ารหัสสามารถให้ค่าอัตราส่วนการบีบอัดที่สูงขึ้นกว่าเมื่อไม่ได้ใช้โครงสร้างต้นไม้นี้ แต่อย่างไรก็ตามการสร้างต้นไม้สำหรับข้อมูลนำเข้า จากนั้นจึงเข้ารหัสนี้ จึงนับเป็นการใช้การอ่านไฟล์ข้อมูลแบบสองครั้ง

ในปี ค.ศ. 2009 โรเบิร์ตและคณะ (Robert et. al.) [15] ใช้การแปลงข้อมูลนำเข้าให้อยู่ในรูปร่างที่เหมาะสม เช่นการแปลงเว้นวรรคให้กลายเป็นสัญลักษณ์อื่น (reducing spaces) การลดสัญลักษณ์ขึ้นบรรทัดใหม่ (end-of-word conversion) เป็นต้น และรายงานว่าการเข้ารหัสเลขคณิตสำหรับคลังข้อมูลแคลการี ให้ค่าอัตราส่วนการบีบอัดที่ดีกว่าเมื่อเปรียบเทียบกับการเข้ารหัสด้วยฮัฟฟ์แมน และลิมเพล-ซิฟ วิธีการแปลงข้อมูลเช่นนี้เหมาะสมอย่างมากกับสัญลักษณ์ในเอกสารภาษาอังกฤษ เนื่องจากเป็นการลดสัญลักษณ์ที่มีความถี่สูงที่สุดคือ เว้นวรรค นั้นออกไปก่อนที่จะเข้ารหัส ดังนั้นผลการทดลองจึงให้ค่าอัตราส่วนการบีบอัดที่สูงมาก แต่ในภาษาอื่นๆ ไม่มีเว้นวรรคระหว่างคำมากเหมือนในเอกสารภาษาอังกฤษ ตัวอย่างเช่น ภาษาไทยเป็นต้น ดังนั้นวิธีการนี้จึงเป็นแนวทางเริ่มต้นที่ดีในการแปลงข้อมูลนำเข้าให้อยู่ในรูปที่เหมาะสมเมื่อใช้กับภาษาอื่น

งานวิจัยทางด้านการประมาณค่าความน่าจะเป็นของสัญลักษณ์นั้นนับเป็นงานวิจัยอีกด้านหนึ่งที่ส่งผลต่อประสิทธิภาพของการเข้ารหัสและถอดรหัสเป็นอย่างมาก ในปี ค.ศ.

1996 บาร์เบอร์และคณะ (Barbir et. al.) [23] ได้นำเสนอการใช้เทคนิคการเข้ารหัสของลิมเพล-ซิฟ เพื่อสร้างเป็นพจนานุกรมของกลุ่มสัญลักษณ์ จากนั้นใช้การเข้ารหัสเลขคณิตที่มีการแปลงความถี่ของสัญลักษณ์ที่คงที่ไปตลอดการเข้ารหัส โดยให้อยู่ในรูปของเลขยกกำลังของสองของกลุ่มสัญลักษณ์นั้นกับคลังข้อมูลแคลการี การใช้ความน่าจะเป็นในรูปแบบของเลขยกกำลังของสองนี้ส่งผลให้การคำนวณในการเข้ารหัสและถอดรหัสเลขคณิตเหลือแต่เพียงการเลื่อนบิตไปทางซ้าย จึงเป็นการลดภาระการคำนวณในการคูณและการหารได้เป็นอย่างดี แต่อย่างไรก็ตามงานวิจัยนี้ไม่ยังได้รวมขนาดของพจนานุกรมที่ส่งผลต่อค่าอัตราส่วนการบีบอัดเข้าไปด้วย

ต่อมาในปี ค.ศ. 2004 คุโรกิและคณะ (Kuroki et. al.) [2] ได้เสนอวิธีการประมาณค่าความน่าจะเป็นของสัญลักษณ์ต่าง ๆ ของไฟล์รูปภาพโดยใช้การกระจายความน่าจะเป็นแบบลาปลาซ (Laplacian distribution) เปรียบเทียบผลกับการเข้ารหัสและถอดรหัสเลขคณิตแบบที่สามารถปรับเปลี่ยนค่าความน่าจะเป็นได้ และรายงานว่าได้รับค่าอัตราส่วนการบีบอัดที่ดีขึ้น 5 เปอร์เซ็นต์

ในปี ค.ศ. 2009 นูเนซ-ยาเนซและคณะ (Nunez-Yanez et. al.) [39] ได้เข้ารหัสกับรูปภาพ โดยใช้แบบจำลองทั้ง 6 แบบของงานวิจัยแชนนอน [30] ในการหาค่าความน่าจะเป็นของสัญลักษณ์ ด้วยการเริ่มต้นจากแบบจำลองลำดับสองในระดับคำและไล่ลงไปหาแบบจำลองลำดับที่ต่ำกว่าในกรณีที่ไม่สามารถหาค่าความน่าจะเป็นในระดับนั้นได้จนกระทั่งหยุดที่แบบจำลองลำดับที่หนึ่ง

บทที่ 3
วิธีดำเนินการวิจัย

3.1 แนวคิดในการดำเนินการวิจัย

ในงานวิจัยนี้ได้ตั้งสมมติฐานไว้ว่า ความน่าจะเป็นของสัญลักษณ์มีส่วนสำคัญอย่างมากต่อค่าอัตราส่วนการบีบอัดของการเข้ารหัสเลขคณิต เนื่องจากค่าความน่าจะเป็น ส่งผลโดยตรงต่อจำนวนบิตรวมตามค่า $\lg(\frac{1}{P(x)})$ นั้น การเตรียมค่าความน่าจะเป็นให้อยู่ในสภาพที่ดีจึงส่งผลต่อค่าอัตราส่วนการบีบอัดของการเข้ารหัสเลขคณิต แต่ใน AAC1 นั้น ค่าความน่าจะเป็นไม่ได้สอดคล้องตามหลักของเอนโทรปีในช่วงแรกของการเข้ารหัส จึงเกิดการผลิตบิตคำตอบที่มีความยาวเกินความจำเป็น ดังนั้นการเตรียมค่าความน่าจะเป็นเริ่มต้นหรือความถี่เริ่มต้นของทุกสัญลักษณ์ ให้อยู่ในสภาพหรือมีความเหมาะสม จึงได้ถูกนำเสนอขึ้นในสองแนวทางด้วยกัน อีกทั้งยังใช้การเข้ารหัสเลขคณิตส่วนเพิ่มขึ้นที่สามารถปรับเปลี่ยนค่าความน่าจะเป็นได้ และกำหนดให้ค่าความน่าจะเป็นเริ่มต้นของสัญลักษณ์ต่างๆ เท่ากัน (incremental adaptive arithmetic coding with equally initial probability of symbol, IAAC1) มาใช้เป็นตัวเปรียบเทียบสำหรับในทุกๆ การทดลองอีกด้วย

3.1.1 การเตรียมค่าความถี่เริ่มต้นโดยการใช้เทคนิคในการจัดกลุ่มค่าความถี่เริ่มต้นของแต่ละสัญลักษณ์ในรหัสแอสกีสำหรับคลังข้อมูลแคลการี

การเตรียมค่าความถี่เริ่มต้นของสัญลักษณ์ในรหัสแอสกีด้วยวิธีนี้ จะใช้เครื่องมือ SPSS 16 ใช้การจัดกลุ่มความถี่ทั้งแบบการใช้ความสัมพันธ์ตามลำดับชั้นและการแบ่งแยกโดยใช้ค่าเฉลี่ยเค โดยมีขั้นตอนหลักตามอัลกอริทึมที่ 3.1 ดังต่อไปนี้

3.1.1.1 การจัดกลุ่มความถี่โดยการใช้ความสัมพันธ์ตามลำดับชั้นในการจัดกลุ่ม (บรรทัดที่ 4 – 26)

- นำไฟล์ข้อความทั้ง 13 ไฟล์มารวมกันเป็นไฟล์เดียว หาค่าความถี่ของแต่ละสัญลักษณ์ ทั้ง 256 สัญลักษณ์แอสกี
- ใช้เทคนิคของการใช้ความสัมพันธ์ตามลำดับชั้นและการแบ่งแยกโดยใช้ค่าเฉลี่ยเค ในการจัดกลุ่ม
- ในกรณีที่เป็นการใช้ความสัมพันธ์ตามลำดับชั้น ให้เลือกจำนวนกลุ่มจากระดับระยะห่างระหว่างกลุ่มที่แปลงค่าแล้วที่มีค่าน้อยที่สุดในเดนโดรแกรม
- ในกรณีที่เป็นการแบ่งแยกโดยใช้ค่าเฉลี่ยเค ให้ใช้จำนวนกลุ่มที่กำหนดขึ้นให้คร่อมกับจำนวนกลุ่มจากเดนโดรแกรม (รวมจำนวนกลุ่มจาก

เดนโดรแกรมด้วย) โดยตัวแปร *method* จะเป็นชนิดของการจัดกลุ่มสองแบบดังกล่าวข้างต้น (บรรทัดที่ 5)

3.1.1.2 การปรับค่าความถี่ของแต่ละสัญลักษณ์ในกลุ่มเดียวกันด้วยค่าเฉลี่ยเลขคณิต (บรรทัดที่ 27 – 31)

- เมื่อได้จำนวนกลุ่มที่เลือกแล้ว แผนภาพเดนโดรแกรมจะแสดงสมาชิกในกลุ่มเดียวกันด้วย ดังนั้นให้นำความถี่ของแต่สัญลักษณ์ในกลุ่มเดียวกันมาหาค่าเฉลี่ยเลขคณิต
- ในกรณีที่เป็นการแบ่งแยกโดยใช้ค่าเฉลี่ยเค ให้ใช้การเป็นสมาชิกภาพของกลุ่ม (cluster membership) มาเป็นตัวบอกว่าความถี่ใดอยู่ในกลุ่มใด จากนั้นให้นำความถี่ของแต่ละสัญลักษณ์ในกลุ่มเดียวกันมาหาค่าเฉลี่ยเลขคณิตเช่นเดียวกัน
- ค่าความถี่ที่เฉลี่ยนี้ ให้ปัดเลขเป็นจำนวนเต็มและเรียกค่าความถี่ ณ จุดนี้ว่า W_2

3.1.1.3 การปรับค่าความถี่เฉลี่ยของทุกกลุ่ม (W_2)ให้เป็นความถี่มาตรฐาน (frequency normalization) (บรรทัดที่ 32 – 40)

- หากลุ่มใน W_2 ที่ให้ค่าความถี่เฉลี่ยที่น้อยที่สุด จากนั้นนำค่าความถี่นี้ไปหารความถี่เฉลี่ยในทุกๆ กลุ่ม
- ค่าความถี่ที่ทำเป็นมาตรฐานนี้ให้ปัดเลขเป็นจำนวนเต็ม และเรียกค่าความถี่ ณ จุดนี้ว่า W_3

เมื่อได้ค่าความถี่ W_3 แล้ว เราจะเรียกความถี่นี้ว่า เป็นค่าความถี่เริ่มต้น M_1 (บรรทัดที่ 41 – 46) เพื่อใช้ในกระบวนการเข้ารหัสเลขคณิตต่อไป อัลกอริทึมที่ 3.1 แสดงถึงการเตรียมค่าความถี่เริ่มต้น M_1

Algorithm 3.1: Construction of M_1

```
1   Input: FILE, clustering_method
2   Output: M1 = [m_{i,1}]_{i=0..256}
3   Begin
4   Let
5         method ← clustering_method
6
7         //set all frequency of symbols to zero
8         For i: 0 ≤ i < 255
9            m_{i,1} = 0;
10        end For
11
```

```
12          //set '<EOF>' frequency to 1
13          m256, 1 ← 1
14
15          //1st Normalization section//
16          //accumulate frequency for each ascii symbol
17          For Each asciiSymbol in FILE
18              mi,1+= asciiSymbol[i];
19          End For
20
21          //Do the hierarchical cluster with
22          D←clustering(mi,1, method)
23
24          //Select appropriate cluster size from
25          //output dendrogram D
26          cs ← appropriateClusterSize(D);
27          //Average the frequency in the same cluster (W2)
28          For k: 0≤ k < cs
29              fav(Dk) ← F (Dk);
30          End For
31
32          //find the minimum of average frequency
33          fav-min(Dk) ← min(fav(Dk))
34
35          //normalize all average cluster frequency of  the minimum
36          //one and ceiling the value of them (W3)
37          For k: 0≤ k < cs
38              fav-norm(Dk) ←⌈fav(Dk)/ fav-min(Dk)⌉;
39          End For
40
41          //for every si in the same cluster
42          //find the fav-norm of si that is in the same cluster  Dk (M1)
43          For i: 0 ≤ i < 255
44              si ∈ Dk ;
45              mi,1 ← fav-norm(Dk);
46          End For
47      End
```

3.1.2 การเตรียมค่าความถี่เริ่มต้นโดยการใช้การกระจายแบบไวบูลล์มาเป็นเครื่องมือในการประมาณค่าความถี่เริ่มต้นของแต่ละสัญลักษณ์ในรหัสแอสกีสำหรับคลังข้อมูลแคนเทอเบอร์รีขนาดใหญ่

ค่าความถี่ของแต่ละสัญลักษณ์ในคลังข้อมูลแคนเทอเบอร์รีขนาดใหญ่ เมื่อนำไปวิเคราะห์หาการกระจายที่เหมาะสมที่สุด จะได้ผลลัพธ์เป็นการกระจายแบบไวบูลล์ เนื่องจากการกระจายแบบไวบูลล์ จะใช้ค่าพารามิเตอร์อยู่สองตัว ตามที่ได้

กล่าวมาแล้วคือ พารามิเตอร์แสดงรูปร่าง (shape parameter - α) และพารามิเตอร์มาตราส่วน (scale parameter - β) ดังนั้นสมมติฐานที่เกี่ยวข้องคือ ค่าพารามิเตอร์แสดงรูปร่าง และค่าพารามิเตอร์มาตราส่วน ควรจะมีค่าเป็นเท่าใดที่จะส่งผลให้เกิดค่าอัตราส่วนการบีบอัดสูงสุดสำหรับการเข้ารหัสเลขคณิตนี้ ภายหลังจากเมื่อเลือกค่าพารามิเตอร์ทั้งสองได้แล้ว เราจึงใช้ฟังก์ชันก่อกำเนิดของการกระจายแบบไวบูลล์หรือสมการ (5) ในการสร้างข้อมูลที่มีการกระจายแบบไวบูลล์จำนวน 256 ตัว ซึ่งรายละเอียดของเตรียมค่าความถี่เริ่มต้นด้วยวิธีการนี้ มีรายละเอียดดังอัลกอริทึมที่ 3.2

Algorithm 3.2: Construction of M_2
1 **Input**: File
2 **Output**: $M_2 = [m_{i,2}]_{i=0..256}$
3 **Begin**
4 F ← calculate the frequency of 256 ascii symbols in FILE
5 F_2← sorting descending order of F
6 DN←fit Distribution with Input Analyzer
7 DN←'Weibull distribution'
8 choose α and β
9 random x and generate data file (D_2) with generating function of $\left[-\frac{1}{\alpha}\ln(1-x)\right]^{\frac{1}{\beta}}$
10 M_2 ← 1-1 Mapping D_2 with F_2
11 **Return** M_2
12 **End**

3.1.2.1 เรียงลำดับค่าความถี่จากค่ามากไปหาค่าน้อย ของแต่ละสัญลักษณ์แอสกีในไฟล์ bible.txt ของคลังข้อมูลแคนเทอเบอร์รีขนาดใหญ่

3.1.2.2 นำค่าความถี่ไปวิเคราะห์หาการกระจายที่เหมาะสมที่สุด (บรรทัดที่ 4–8)

- หาการกระจายที่เหมาะสมที่สุด โดยการใช้โปรแกรม Input Analyzer 11 ของโปรแกรม ARENA
- ให้เลือกการกระจายในอันดับแรก ซึ่งในที่นี้เป็นเลือกการกระจายแบบไวบูลล์เป็นต้นแบบในการคิดหาค่าความถี่เริ่มต้น
- เลือกค่าพารามิเตอร์แสดงรูปร่าง และพารามิเตอร์มาตราส่วน โดยการเลือกค่าพารามิเตอร์แสดงรูปร่างควรจะมีค่าไม่เกิน 1

3.1.2.3 ใช้ฟังก์ชันก่อกำเนิดของการกระจายแบบไวบูลล์ในการสร้างข้อมูลจำนวน 256 ตัว (บรรทัดที่ 9)

3.1.2.4 นำข้อมูลจำนวน 256 ตัวที่ได้ไปเรียงลำดับตามค่าจากมากไปหาค่าน้อย และจับคู่แบบหนึ่งต่อหนึ่ง กับสัญลักษณ์ในข้อ 3.1.2.1 (บรรทัดที่ 10)

3.1.2.5 เราจะเรียกค่าของตัวเลขในข้อ 3.1.2.4 นี้ว่าค่าความถี่เริ่มต้น M_2 (บรรทัดที่ 11)

นอกจากการใช้เทคนิคการเตรียมค่าความถี่เริ่มต้นทั้งสองวิธีที่ได้กล่าวมาดังข้างต้น เรายังสามารถใช้เทคนิคอื่นๆ ร่วมด้วยคือ การลดความแตกต่างของค่าความน่าจะเป็นของสัญลักษณ์ที่มีความถี่สูงสุดและความถี่ต่ำสุดให้น้อยลง การลดทอนสัญลักษณ์ที่ไม่ปรากฏ สุดท้ายเป็นการพิจารณาการแบ่งไฟล์ออกเป็นส่วนๆ เมื่อพบว่าค่าอัตราส่วนการบีบอัดในขณะนั้นมีค่าคงที่หรือมีค่าลดลง โดยรายละเอียดในแต่ละวิธีมีดังต่อไปนี้

3.1.3 การลดความแตกต่างของค่าความน่าจะเป็นของสัญลักษณ์ที่มีความถี่สูงสุดและความถี่ต่ำสุดให้น้อยลง

ค่าความน่าจะเป็นหรือค่าความถี่ของสัญลักษณ์ที่ได้จากอัลกอริทึมที่ 3.1 ภายหลังจากการปรับค่าความถี่เฉลี่ยของทุกกลุ่ม (W_2) ให้เป็นความถี่มาตรฐานแล้ว เรายังสามารถลดขนาดของจำนวนเท่าของค่าความถี่ของสัญลักษณ์ของกลุ่มที่มีค่าความน่าจะเป็นสูง ต่อค่าความถี่ของสัญลักษณ์ของกลุ่มที่มีค่าต่ำด้วยการหารด้วยเลขจำนวนเต็ม $\mathbb{N}$ โดยที่ $\mathbb{N} \geq 2$ ซึ่งเมื่อนำจำนวนเต็ม $\mathbb{N}$ นี้ไปหารค่าความถี่นี้แล้ว จะทำให้ค่าความถี่ของสัญลักษณ์ในกลุ่มที่มีค่าความถี่ต่ำสุดมีค่าน้อยกว่า 1 และให้ปัดเศษขึ้นในทุกๆ ผลลัพธ์ของการหาร ดังนั้นวิธีนี้ค่าความถี่ของสัญลักษณ์ในกลุ่มที่ต่ำจะมีค่าต่ำที่สุดเป็น 1 ตลอด ในขณะที่ความถี่ของสัญลักษณ์ของกลุ่มที่มีความถี่สูงจะมีค่าลดลงเรื่อยๆ จนกระทั่งไม่มีผลต่อการเพิ่มขึ้นของค่าอัตราส่วนการบีบอัดของการเข้ารหัสเลขคณิต โดยค่าอัตราส่วนการบีบอัดนี้สามารถอธิบายได้ดังนี้

กำหนดให้

f_{HB} แทนความถี่ของกลุ่มสัญลักษณ์ความถี่สูง ก่อนการหารด้วย $\mathbb{N}$

f_{LB} แทนความถี่ของกลุ่มสัญลักษณ์ความถี่ต่ำสุด ก่อนการหารด้วย $\mathbb{N}$

f_{hA} แทนความถี่ของกลุ่มสัญลักษณ์ความถี่สูง หลังการหารด้วย $\mathbb{N}$

f_{lA} แทนความถี่ของกลุ่มสัญลักษณ์ความถี่ต่ำสุด หลังการหารด้วย $\mathbb{N}$

n_B แทนจำนวนตัวของสัญลักษณ์ทั้งหมด ก่อนการหารด้วย $\mathbb{N}$

n_A แทนจำนวนตัวของสัญลักษณ์ทั้งหมด หลังการหารด้วย $\mathbb{N}$

P_{hA} แทนความน่าจะเป็นของกลุ่มสัญลักษณ์ความถี่สูง
P_{lA} แทนความน่าจะเป็นของกลุ่มสัญลักษณ์ความถี่ต่ำ

ดังนั้น

$$f_{hA} = \left\lceil \frac{f_{HB}}{\mathbb{N}} \right\rceil, f_{lA} = \left\lceil \frac{f_{LB}}{\mathbb{N}} \right\rceil, n_A = \frac{n_B}{\mathbb{N}}$$

แต่

$$P_{hA} = \frac{f_{hA}}{n_A} = \frac{\left\lceil \frac{f_{HB}}{\mathbb{N}} \right\rceil}{\frac{n_B}{\mathbb{N}}} \geq \frac{f_{HB}}{n_B} \geq P_{HB}$$

และ

$$P_{lA} = \left\lceil \frac{f_{lA}}{n_A} \right\rceil = \left\lceil \frac{f_{lB}}{n_B} \right\rceil = 1 = P_{LB}$$

จะได้ว่า

$$-\sum(\lg(P_{hA}) + \lg(P_{lA})) \leq -\sum(\lg(P_{HB}) + \lg(P_{LB}))$$

หรือ

จำนวนบิตของบิตคำตอบหลังการหารความน่าจะเป็นเริ่มต้นด้วยค่า $\mathbb{N}$ มีค่าน้อยกว่าจำนวนบิตของบิตคำตอบก่อนการหารความน่าจะเป็นเริ่มต้นด้วยค่า $\mathbb{N}$

3.1.4 การลดทอนสัญลักษณ์ที่ไม่ปรากฏ

สัญลักษณ์ในรหัสแอสกี ได้ถูกนำมาใช้ในเอกสารภาษาอังกฤษเพียงบางส่วน โดยเฉพาะอย่างยิ่ง ไฟล์ประเภทข้อความด้วยแล้ว เช่น กลุ่มของสัญลักษณ์ในรหัสแอสกีตั้งแต่ 128 ขึ้นไป หรือกลุ่มรหัสแอสกีที่ต่ำกว่า 10 ลงมา แต่ใน IAAC1 นั้น เริ่มต้นให้ทุกสัญลักษณ์มีค่าความน่าจะเป็นเท่ากัน การทำให้ความน่าจะเป็นของสัญลักษณ์ที่ไม่ได้ใช้บางตัวให้มีค่าเป็น 0 จะทำให้ค่าความน่าจะเป็นของสัญลักษณ์ที่ใช้ตัวอื่นๆ มีค่าความน่าจะเป็นมากขึ้น ส่งผลให้จำนวนบิตของคำตอบที่ได้มีค่าน้อยลง นั่นย่อมทำให้ค่าอัตราส่วนการบีบอัดสูงขึ้นตามไปด้วย เป็นที่สังเกตว่าสัญลักษณ์ที่ไม่ปรากฏอยู่นำเอกสารภาษาอังกฤษนี้ ส่วนใหญ่แล้วเป็นสัญลักษณ์ที่จัดว่ามีค่าความน่าจะเป็นน้อยที่สุด เมื่อเทียบกับกลุ่มอื่นๆ ที่มีความน่าจะเป็นสูงกว่า โดยการลดทอนสัญลักษณ์ที่ไม่ปรากฏนี้ สามารถเพิ่มค่าอัตราส่วนการบีบอัดที่สามารถอธิบายได้ดังนี้

กำหนดให้

$(P_B)_i$ แทนความน่าจะเป็นของสัญลักษณ์ที่ i ใดๆ ก่อนการลดทอนสัญลักษณ์ที่มีค่า > 0

$(P_A)_i$ แทนความน่าจะเป็นของสัญลักษณ์ที่ i ใดๆ หลังการลดทอนสัญลักษณ์ที่มีค่า > 0

เนื่องจาก

$$(P_A)_i \geq (P_B)_i$$

ดังนั้น

$$-\sum \lg(P_A)_i \leq -\sum \lg(P_B)_i$$

หรือ

จำนวนบิตคำตอบของสัญลักษณ์ที่ได้รับความน่าจะเป็นเริ่มต้นเพิ่มขึ้นจะมีค่าน้อยกว่าเมื่อสัญลักษณ์นั้นมีความน่าจะเป็นเริ่มต้นน้อยกว่า

3.1.5 การพิจารณาการแบ่งไฟล์ออกเป็นส่วน ๆ เมื่อพบว่าค่าอัตราส่วนการบีบอัดในขณะนั้นมีค่าคงที่หรือมีค่าลดลง

การพิจารณาการแบ่งไฟล์ออกเป็นส่วนๆ นี้ มีแนวคิดในการพิจารณาหาจุดแบ่งได้ออกเป็นสองหัวข้อได้แก่ การแบ่งไฟล์ด้วยค่าขีดจำกัด-ค่าตัวคูณ (threshold-factor) โดยใช้ค่าอัตราส่วนการบีบอัดที่เปลี่ยนแปลงไป $|\Delta CR|$ หรือค่าอัตราส่วนการบีบอัดเฉพาะด้านบวกเป็นเกณฑ์ และการแบ่งไฟล์ด้วยการพิจารณาปัจจัยที่ส่งผลต่อค่า ΔCR ทั้งค่าที่เป็นลบ และค่าที่เป็นบวกเป็นเกณฑ์ โดยมีรายละเอียดในแต่ละหัวข้อดังต่อไปนี้

3.1.5.1 การแบ่งไฟล์ด้วยค่าขีดจำกัด-ค่าตัวคูณ (threshold-factor: $\tau - \varphi$) โดยใช้ค่าอัตราส่วนการบีบอัดที่เปลี่ยนแปลงไปของ $|\Delta CR|$ เป็นเกณฑ์

ในทุกรอบของการเข้ารหัสเลขคณิต ถ้ารอบใดมีจำนวนของบิตคำตอบสะสมเท่ากับ 8 บิตแล้ว (1 byte) ให้เราคำนวณค่าอัตราส่วนการบีบอัด และค่าอัตราส่วนการบีบอัดที่เปลี่ยนแปลงไป ณ รอบนั้น เมื่อพิจารณาค่าสัมบูรณ์ของค่าอัตราส่วนการบีบอัดที่เปลี่ยนแปลงไป ถ้ามีขนาดน้อยกว่าค่าขีดจำกัดที่ตั้งไว้ค่าหนึ่ง (τ: $\tau < 1$) ให้ถือจุดนั้นเป็นจุดแบ่งไฟล์ ภายหลังจากจุดนั้นแล้วให้เริ่มต้นการเข้ารหัสเลขคณิตด้วยความถี่เริ่มต้นที่ได้จากอัลกอริทึมที่ 3.1 หรือ 3.2 อย่างใดอย่างหนึ่งใหม่ อีกทั้งมีการลดขนาดของค่าขีดจำกัดลงด้วยค่าตัวคูณ (φ: $0 <$

$\varphi < 1$) เพื่อทำให้ค่าขีดจำกัดที่ได้ใหม่มีค่าน้อยลง และสอดคล้องกับค่าอัตราส่วนการบีบอัดที่เปลี่ยนแปลงไปที่จะมีค่าน้อยลงเช่นกัน อีกทั้งเพื่อไม่ทำให้เกิดการแบ่งไฟล์หลายรอบมากจนเกินไป ค่าจำนวนของบิตคำตอบสะสม และจำนวนบิตที่อ่านเข้าไปสะสมจะมีค่าเพิ่มมากขึ้น จนกระทั่งจำนวนบิตคำตอบสะสมสุดท้ายมีค่าเท่ากับจำนวนของบิตคำตอบทั้งหมดที่ได้จากการเข้ารหัส และจำนวนบิตที่อ่านเข้าไปสะสมสุดท้ายมีค่าเท่ากับขนาดของไฟล์เริ่มต้นตามลำดับ อัลกอริทึมที่ 3.3 แสดงการหาจุดแบ่งไฟล์โดยการวิเคราะห์ค่าอัตราส่วนการบีบอัดที่เปลี่ยนแปลงไป เมื่อเทียบกับค่าขีดจำกัดที่ตั้งไว้

Algorithm 3.3: isPartition

```
1   Input:    threshold (τ) , factor (φ), |bitin|, |bitout|, M1
2   Output:   isPartition
3   Begin
4   Let       isParition← false
5             CR = |bitin| / |bitout|
6             If (|ΔCR | < τ)
7                 τ ← τ × φ
8                 isPartition ← true
9                 M ← M1
10            End If
11  Return isPartition
12  End
```

บรรทัดที่ 4 กำหนดตัวแปรบูลีน *isPartition* ให้มีค่าเป็นเท็จ

บรรทัดที่ 5 เป็นการคิดค่าอัตราส่วนการบีบอัดจากจำนวนของ $|bitin|, |bitout|$

บรรทัดที่ 6 - 10 คำนวณค่าอัตราส่วนการบีบอัดที่เปลี่ยนแปลง จากการคิดค่าอัตราส่วนการบีบอัดในรอบก่อนหน้า เทียบกับในรอบปัจจุบัน เมื่อมีค่าน้อยกว่าค่าขีดจำกัดที่ตั้งไว้ ให้คำนวณค่าขีดจำกัดใหม่ จากค่าตัวคูณ (บรรทัดที่ 7) และกำหนดตัวแปรบูลีน *isPartition* ให้มีค่าเป็นจริง (บรรทัดที่ 8) จากนั้นให้เปลี่ยนค่าถี่ของสัญลักษณ์ในขณะนั้น ให้กลายเป็นค่าถี่เริ่มต้นของสัญลักษณ์แทน (บรรทัดที่ 9)

3.1.5.2 การแบ่งไฟล์ด้วยค่าขีดจำกัด-ค่าตัวคูณ (threshold-factor: $\tau - \varphi$) โดยใช้ค่าอัตราส่วนการบีบอัดที่เปลี่ยนแปลงไปของ ΔCR เป็นเกณฑ์

จากเทคนิคในหัวข้อ 3.1.5.1 พบว่าค่าขีดจำกัดและค่าตัวคูณสามารถเพิ่มค่าอัตราส่วนการบีบอัดได้ เป็นเพียงการพิจารณาเฉพาะ ΔCR ที่เป็นบวกอย่างเดียว นั่นหมายถึงว่า ณ ตำแหน่งที่ค่าอัตราส่วนการบีบอัดไม่ได้ดีขึ้นไปกว่าค่าขีดจำกัดที่ตั้งไว้ จะถือว่าให้ตำแหน่งนั้นเป็นจุดในการแบ่งไฟล์ ซึ่งในความเป็นจริงแล้วค่า ΔCR ที่ติดลบไม่ได้รับการพิจารณาเพื่อเป็นเกณฑ์ในการแบ่งไฟล์ด้วย ดังนั้นแนวทางในการปรับปรุงเกณฑ์ดังกล่าวนั้นจึงได้ปรับให้เป็นไปดังอัลกอริทึมที่ 3.4 โดยการพิจารณาทั้งค่า ΔCR ที่เป็นบวก (ΔCR^{+}) และ ΔCR ที่เป็นลบ (ΔCR^{-}) นอกจากนั้นยังได้เพิ่มการพิจารณาค่าการลดลงอย่างต่อเนื่องของค่า ΔCR (δ^{-}) ถ้า ณ ตำแหน่งใดมีค่าสะสมของค่า δ^{-} ที่ค่าน้อยกว่าค่าจำนวนลบติดต่อกันสูงสุด (δ^{*}) ให้ถือว่า ณ ตำแหน่งนั้นเป็นจุดแบ่งไฟล์ได้อีกด้วย ดังแสดงได้ในอัลกอริทึม 3.4

Algorithm 3.4: isNewPartition

Input: *threshold* (τ) , *factor* (φ), $|bitin|$, $|bitout|$, M_1
Output: *isPartition*
Begin
Let *isParition*← false
$CR = \frac{|bitin|}{|bitout|}$
If ($\Delta CR^{-} < 0$)
$\delta^{-} \leftarrow \delta^{-} + 1$
If ($\delta^{-} > \delta^{*}$)
$\delta^{-} \leftarrow 0,\ |bitin| \leftarrow 0, |bitout| \leftarrow 0$
$M \leftarrow M_1$
isPartition ← true
End If
else
$\delta^{-} \leftarrow 0$
If($\Delta CR^{+} < \tau$)
$\tau \leftarrow \tau \times \varphi$
$|bitin| \leftarrow 0, |bitout| \leftarrow 0$
$M \leftarrow M_1$
isPartition ← true
End If
End If
Return *isPartition*
End

บรรทัดที่ 4 กำหนดตัวแปรบูลีน *isPartition* ให้มีค่าเป็นเท็จ

บรรทัดที่ 5 เป็นการคิดค่าอัตราส่วนการบีบอัดจากจำนวนของ $|bitin|, |bitout|$

บรรทัดที่ 6 - 22 คำนวณค่าอัตราส่วนการบีบอัดที่เปลี่ยนแปลง จากการคิดค่าอัตราส่วนการบีบอัดในรอบก่อนหน้า เทียบกับในรอบปัจจุบัน และแยกออกเป็นสองกรณีคือ

ในกรณีที่ $\Delta CR < 0$

บรรทัดที่ 7 ให้นับจำนวน δ^- เพิ่มขึ้น 1

บรรทัดที่ 15 - 20 ถ้าจำนวน δ^- มีค่ามากกว่าจำนวน δ^* ให้กำหนดให้ δ^-, $|bitin|$, $|bitout|$ มีค่าเป็น 0 (บรรทัดที่ 16) จากนั้นให้เปลี่ยนค่าความถี่ของสัญลักษณ์ในขณะนั้น ให้กลายเป็นค่าถี่เริ่มต้น (บรรทัดที่ 18) และกำหนดตัวแปรบูลีน *isPartition* ให้มีค่าเป็นจริง (บรรทัดที่ 11)

ในกรณีที่ $\Delta CR > 0$

บรรทัดที่ 14 ให้ δ^- มีค่าเป็น 0

บรรทัดที่ 15 - 20 ถ้า ΔCR มีค่าน้อยกว่าขีดจำกัดที่ตั้งไว้ ให้คำนวณค่าขีดจำกัดใหม่ จากค่าตัวคูณ (บรรทัดที่ 16) กำหนดให้ $|bitin|$, $|bitout|$ มีค่าเป็น 0 (บรรทัดที่ 17) จากนั้นให้เปลี่ยนค่าถี่ของสัญลักษณ์ในขณะนั้น ให้กลายเป็นค่าความถี่เริ่มต้น (บรรทัดที่ 18) และกำหนดตัวแปรบูลีน *isPartition* ให้มีค่าเป็นจริง (บรรทัดที่ 19)

3.1.6 การหาค่าพารามิเตอร์แสดงรูปร่างและพารามิเตอร์มาตราส่วนของการกระจายแบบไวบูลล์ (α, β) ที่ให้ค่าอัตราส่วนการบีบอัดของเอกสารภาษาอังกฤษที่สูงสุด

ในการกำหนดค่าความถี่เริ่มต้นตามอัลกอริทึม 3.2 นั้น ใช้บรรทัดที่ 9 และ 10 เป็นหลัก โดยเป็นการสุ่มตัวเลขที่สอดคล้องกับการกระจายแบบไวบูลล์ ตามค่าพารามิเตอร์ทั้งสองที่กำหนดลงไป ซึ่งในบางครั้งค่าความถี่เริ่มต้นที่ได้ สามารถที่ใช้เทคนิคในหัวข้อที่ 3.1.3 การลดความแตกต่างของค่าความน่าจะเป็นของสัญลักษณ์ที่มีความถี่สูงสุดและความถี่ต่ำสุด ให้แตกต่างน้อยลงไปได้อีก เพื่อให้ค่าอัตราส่วนการบีบอัดที่ได้มีค่าสูงสุด เราจึงจำเป็นที่จะต้องมีการสร้างความถี่เริ่มต้นหลายครั้งด้วยกัน เพื่อรับประกันความถูกต้องดังในอัลกอริทึมที่ 3.5 โดยอัลกอริทึมนี้เป็นการหาค่าพารามิเตอร์แสดงรูปร่างและพารามิเตอร์มาตราส่วนที่ระดับทศนิยมตำแหน่งที่หนึ่ง ที่ทำให้ค่าอัตราส่วนการบีบอัดของกลุ่มของไฟล์ข้อความที่กำหนดไว้ได้ค่าสูงที่สุด

Algorithm 3.5: Determine the value of α_{CRmax} and β_{CRmax}

```
1   Input:   FILE, α_min , α_max ,α_step, β_min, β_max and β_step
2   Output:  α_CRmax, β_CRmax and CR_max, M_α β
3   Begin
4   Let   CR_max = 0
5       For(i = 0;  i ≤ 30; i++)
6         For (α = α_min ; α ≤ α_max; α += α_step)
7          For (β = β_min; β ≤ β_max; β += β_step)
8            ▪ random x.
9            ▪ Generate data y from −β[ln(1 − x)]^(1/α)
10           ▪ Sort y in ascending order.
11           ▪ Mapping y (1:1) with the ascending frequency
               order of symbol. (M_α β)
12           ▪ Elimination of unused symbols.
13           ▪ Incremental adaptive arithmetic coding (IAAC)
14           ▪ Calculate CR_α,β.
15             If(CR_α,β > CR_max)
16                CR_max ← CR_α,β
17                α_CRmax ← α
18                β_CRmax ← β
19             End If
20          End For
21        End For
22      End For
23  Return CR_max, M_α β, α_CRmax, β_CRmax
24  End
```

บรรทัดที่ 4 กำหนดให้ค่า CR_{max} มีค่าเท่ากับศูนย์

บรรทัดที่ 5 ให้ทำการหาค่า CR ของกลุ่มของไฟล์ที่กำหนดจำนวน 30 ครั้ง

บรรทัดที่ 6 วนลูปค่าพารามิเตอร์แสดงรูปร่าง α ด้วยค่า α_{step} จากค่า α_{min} จนถึงค่า α_{max}

บรรทัดที่ 7 วนลูปค่าพารามิเตอร์มาตราส่วน β ด้วยค่า β_{step} จากค่า β_{min} จนถึงค่า β_{max}

บรรทัดที่ 8 สุ่มค่าตัวแปรสุ่ม x

บรรทัดที่ 9 จากค่า α, β และ x ให้คำนวณหาข้อมูล y ที่สอดคล้องกับ $-\beta[\ln(1-x)]^{\frac{1}{\alpha}}$

บรรทัดที่ 10 เรียงลำดับค่า y จากค่ามากไปหาค่าน้อย

บรรทัดที่ 11 จับคู่ค่าของ y กับค่าความถี่ที่เรียงจากค่ามากไปหาค่าน้อยของสัญลักษณ์ที่ได้จากไฟล์ *bible.txt*

บรรทัดที่ 12 การลดทอนสัญลักษณ์ที่ไม่ปรากฏ

บรรทัดที่ 13 นำค่าความถี่เริ่มต้นที่จากบรรทัดที่ 12 มาเข้ารหัสเลขคณิตส่วนเพิ่มขึ้นที่สามารถปรับเปลี่ยนค่าความน่าจะเป็นได้

บรรทัดที่ 14 คำนวณหาค่าอัตราส่วนการบีบอัดของแต่ละไฟล์ข้อความ

บรรทัดที่ 15 – 19 ถ้าค่าอัตราส่วนการบีบอัดเฉลี่ยของไฟล์ข้อความที่กำหนดให้มีค่ามากกว่าค่า CR_{max} ให้กำหนดให้ CR_{max} มีค่าเท่ากับค่า CR ปัจจุบัน (บรรทัดที่ 16) กำหนดให้ค่า α_{CRmax} มีค่าเท่ากับ α ในรอบปัจจุบัน (บรรทัดที่ 17) และกำหนดให้ค่า β_{CRmax} มีค่าเท่ากับ β ในรอบปัจจุบัน (บรรทัดที่ 18)

บรรทัดที่ 23 ส่งคืนค่าอัตราส่วนการบีบอัดสูงสุด (CR_{max}), ค่าถี่เริ่มต้นที่ได้จากค่า α_{CRmax} และ β_{CRmax} ($M_{\alpha,\beta}$) , ค่าพารามิเตอร์แสดงรูปร่างและค่าพารามิเตอร์มาตราส่วนที่ให้ค่าอัตราส่วนการบีบอัดสูงสุด (α_{CRmax}, β_{CRmax}) ตามลำดับ

บทที่ 4
ผลการวิเคราะห์ข้อมูล

4.1 รายละเอียดคลังข้อมูล

คลังข้อมูลที่ใช้ในงานวิจัยครั้งนี้ ประกอบด้วยคลังข้อมูลแคลการี (Calgary corpus) [1, 15, 17-18, 23] คลังข้อมูลแคนเทอเบอร์รีขนาดใหญ่ (Large Canterbury corpus) [1, 5, 17, 23, 27] กลุ่มของไฟล์ข้อความที่ได้เตรียมไว้ เพื่อทวนสอบประสิทธิภาพของอัลกอริทึมที่นำเสนอ ได้แสดงดังในตารางที่ 4.1 รายละเอียดทั่วไปของคลังข้อมูลแคลการีจำนวน 18 ไฟล์ (นำมาใช้จริงเพียง 13 ไฟล์ เนื่องจากเป็นไฟล์ข้อความ) ตารางที่ 4.2 รายละเอียดทั่วไปของคลังข้อมูลแคนเทอเบอร์รีขนาดใหญ่จำนวน 3 ไฟล์ และตารางที่ 4.3 รายละเอียดทั่วไปของกลุ่มของไฟล์นิทานภาษาอังกฤษ ซึ่งจัดเป็นเอกสารข้อความภาษาอังกฤษจำนวน 9 ไฟล์ตามลำดับ

ตารางที่ 4.1 รายละเอียดทั่วไปของคลังข้อมูลแคลการี

ลำดับที่	ชื่อไฟล์	ขนาดของไฟล์ (KB)	ประเภทของไฟล์	นำมาใช้
1	bib	111,262	ข้อมูลในการใช้อ้างอิงการเขียนบทความ (bibliographic data)	☑
2	book1	768,771	ข้อมูลเกี่ยวกับหนังสือ ผู้แต่ง ฯลฯ	☑
3	book2	610,856		☑
4	geo	102,400	ไฟล์ไบนารี (Binary file)	☐
5	news	377,109	ข้อความโต้ตอบในอีเมล	☑
6	obj1	21,504	ไฟล์ไบนารี	☐
7	obj2	246,814		☐
8	paper1	53,161	ข้อมูลที่รวมถึงโครงสร้างในการจัดเก็บบทความทางวิชาการ ชื่อ และบทคัดย่อ ฯลฯ	☑
9	paper2	82,199		☑
10	paper3	46,526		☑
11	paper4	13,286		☑
12	paper5	11,954		☑
13	paper6	38,105		☑
14	pic	513,216	ไฟล์ไบนารี	☐

15	progc	39,611	ภาษาในการเขียนโปรแกรม (programming language)	☑
16	progl	71,646		☑
17	progp	49,379		☑
18	trans	93,695	ข้อความที่มีโครงสร้างเฉพาะ	☐

ตารางที่ 4.2 รายละเอียดทั่วไปของคลังข้อมูลแคนเทอเบอร์รีขนาดใหญ่

ลำดับที่	ชื่อไฟล์	ขนาดของไฟล์ (KB)	ประเภทของไฟล์	นำมาใช้
1	bible.txt	4,047,392	ข้อมูลในไบเบิ้ล	☑
2	e.coli	4,638,690	ข้อมูลการเรียงตัวของสายดีเอ็นเอ	☑
3	world.txt	2,473,400	ข้อมูลเบื้องต้นเกี่ยวกับประเทศในโลก	☑

ตารางที่ 4.3 รายละเอียดทั่วไปของกลุ่มของไฟล์นิทานภาษาอังกฤษ

ลำดับที่	ชื่อไฟล์	ขนาดของไฟล์ (KB)	ประเภทของไฟล์	นำมาใช้
1	A Tale of Two Cities	776,629	นิทานภาษาอังกฤษทั่วไป	☑
2	Ethan Frome	203,305		☑
3	Heart of Darkness	229,831		☑
4	Moby Dick	1,231,973		☑
5	Native Son	32,020		☑
6	Pride and Prejudice	704,158		☑
7	Robinson Crusoe	642,573		☑
8	Silas Marner	413,529		☑
9	The Invisible Man	292,663		☑

4.2 การออกแบบการทดลอง

การออกแบบการทดลองในบทนี้ ประกอบด้วยการทดลองหลัก 5 การทดลองดังต่อไปนี้

4.2.1 การปรับปรุงค่าอัตราส่วนการบีบอัดของคลังข้อมูลแคลการี โดยการใช้เทคนิคของการจัดกลุ่มแบบการใช้ความสัมพันธ์ตามลำดับชั้นมาใช้ในการเตรียมค่าความถี่เริ่มต้น

ในการทดลองนี้ใช้ไฟล์จากคลังข้อมูลแคลการีจำนวน 13 ไฟล์คือ bib, book1, book2, news, paper1, paper2, paper3, paper4, paper5, paper6, progc, progl และ progp จากนั้นใช้อัลกอริทึมที่ 3.1 ในการสร้างค่าความถี่เริ่มต้นในการเข้ารหัสเลขคณิตส่วนเพิ่มขึ้นที่สามารถปรับเปลี่ยนค่าความน่าจะเป็นได้ อีกทั้งใช้เทคนิคในการจัดกลุ่ม แบบการใช้ความสัมพันธ์ตามลำดับชั้น(Interval: Euclidean distance, Cluster method: nearest neighbor) ที่ใช้ผลรวมของความถี่ของสัญลักษณ์ของทั้ง 13 ไฟล์นั้นมาเป็นข้อมูลนำเข้า พร้อมทั้งใช้เทคนิคในหัวข้อ 3.1.3 – 3.1.5 ได้แก่ การลดความแตกต่างของค่าความน่าจะเป็นของ ที่มีความถี่สูงสุดและความถี่ต่ำสุดให้น้อยลง โดยใช้ $\mathbb{N} \in \{1,2,3,4,5,6,7\}$ รวมถึงการลดทอนสัญลักษณ์ที่ไม่ปรากฏและการพิจารณาการแบ่งไฟล์ออกเป็นส่วนๆ เมื่อพบว่าค่าอัตราส่วนการบีบอัดในขณะนั้นมีค่าคงที่หรือมีค่าลดลง เปรียบเทียบค่าขนาดของไฟล์และค่าอัตราส่วนการบีบอัดเทียบกับ IAAC1

4.2.2 การปรับปรุงค่าอัตราส่วนการบีบอัดของคลังข้อมูลแคลการี โดยการใช้เทคนิคของการจัดกลุ่มแบบการแบ่งแยกโดยใช้ค่าเฉลี่ยเคมาใช้ในการเตรียมค่าความถี่เริ่มต้น

การทดลองนี้ต่างจากการทดลองที่แล้วเพียงแต่เปลี่ยนวิธีในการจัดกลุ่ม เป็นการแบ่งแยกโดยใช้ค่าเฉลี่ยเคแทน โดยจำนวนกลุ่มหรือค่า k จะมีค่าเท่ากับ 5, 8, 11, 14 และ 17 กลุ่ม นำเสนอเฉพาะผลของการเข้ารหัสที่ให้ค่าอัตราส่วนการบีบอัดสูงที่สุดเพียงกลุ่มเดียว จากนั้นเปรียบเทียบค่านี้กับผลการทดลองในหัวข้อ 4.2.1

4.2.3 การปรับปรุงค่าอัตราส่วนการบีบอัดของกลุ่มของไฟล์นิทานภาษาอังกฤษ

การทดลองนี้ใช้อัลกอริทึมที่ 3.5 ในการหาค่า α_{CRmax} และ β_{CRmax} ของไฟล์นิทานภาษาอังกฤษทุกไฟล์ ได้แสดงดังในตารางที่ 4.3 ได้แก่ A Tale of Two Cities, Ethan Frome, Heart of Darkness, Moby Dick, Native Son, Pride and Prejudice, Robinson Crusoe, Silas Marner และ The Invisible Man ก่อนที่จะส่งค่าดังกล่าวไปเข้ารหัส โดยใช้อัลกอริทึมที่ 3.2 ในการสร้างค่าความถี่เริ่มต้น โดยนำค่า α_{CRmax} และ β_{CRmax} ที่ได้มาใช้ใน

การเข้ารหัสเลขคณิตส่วนเพิ่มขึ้นที่สามารถปรับเปลี่ยนค่าความน่าจะเป็นได้ กับไฟล์นิทานภาษาอังกฤษทุกไฟล์ ตามตารางที่ 4.3 เปรียบเทียบค่าขนาดของไฟล์และค่าอัตราส่วนการบีบอัดเทียบกับ IAAC1

4.2.4 การเปรียบเทียบเทคนิคในการเตรียมค่าความถี่เริ่มต้น

การทดลองนี้จะเปรียบเทียบเทคนิคทั้งหมดเพื่อเป็นการทวนสอบประสิทธิผลของอัลกอริทึม (Validation) ทั้งเทคนิคการเตรียมค่าความถี่เริ่มต้นด้วยเทคนิคในการจัดกลุ่มแบบการใช้ความสัมพันธ์ตามลำดับชั้น (หัวข้อ 4.2.1) เทคนิคในการจัดกลุ่มแบบการแบ่งแยกโดยใช้ค่าเฉลี่ยเค ที่จำนวนกลุ่มที่ให้ค่าอัตราส่วนการบีบอัดสูงสุด (หัวข้อ 4.2.2) และเทคนิคในการประมาณค่าความถี่เริ่มต้นที่ได้จากการกระจายแบบไวบูลล์ ที่ใช้ค่า α_{CRmax} และ β_{CRmax} จากหัวข้อ 4.2.3 มาเข้ารหัสเลขคณิตส่วนเพิ่มขึ้นที่สามารถปรับเปลี่ยนค่าความน่าจะเป็นได้ กับคลังข้อมูลแคนเทอเบอร์รีขนาดใหญ่ซึ่งแสดงดังในตารางที่ 4.11 และ 4.12 อีกทั้งยังเปรียบเทียบกับค่าอัตราส่วนการบีบอัดของ IAAC1 ด้วย

การเปรียบเทียบอีกรูปแบบหนึ่ง มีวัตถุประสงค์เพื่อเปรียบเทียบประสิทธิผลของการใช้ค่าความถี่เริ่มต้นของสัญลักษณ์ ทั้งเทคนิคการเตรียมค่าความถี่เริ่มต้นด้วยเทคนิคในการจัดกลุ่ม แบบการใช้ความสัมพันธ์ตามลำดับชั้น (หัวข้อ 4.2.1) เทคนิคในการจัดกลุ่มแบบการแบ่งแยกโดยใช้ค่าเฉลี่ยเค ที่จำนวนกลุ่มที่ให้ค่าอัตราส่วนการบีบอัดสูงสุด (หัวข้อ 4.2.2) ซึ่งเป็นความถี่เริ่มต้นของไฟล์ข้อความทั่วไป มาเปรียบเทียบกับค่าอัตราส่วนการบีบอัดของไฟล์นิทานภาษาอังกฤษ (ตารางที่ 4.3) ที่ใช้ความถี่เริ่มต้นจากการกระจายแบบไวบูลล์ เปรียบเทียบผลลัพธ์จากตารางที่ 4.9 ดังแสดงไว้ในตารางที่ 4.13 และ 4.14

4.2.5 การทดสอบความสำคัญของค่าความถี่เริ่มต้นของสัญลักษณ์

การทดสอบนี้มีจุดประสงค์เพื่อศึกษาถึง ความสำคัญของค่าความถี่เริ่มต้นของสัญลักษณ์ที่ได้จากเทคนิคการเตรียมค่าความถี่เริ่มต้นด้วยเทคนิคในการจัดกลุ่ม แบบการใช้ความสัมพันธ์ตามลำดับชั้น (τ = 0.001, φ = 0.04, $\delta^* = 15$, Gap-4) และเทคนิคในการจัดกลุ่มแบบการแบ่งแยกโดยใช้ค่าเฉลี่ยเค (τ = 0.001, φ = 0.04, $\delta^* = 15$, k = 14, Gap-6) จากคลังข้อมูลแคลการีเพื่อนำไปสรุปหาลักษณะของความถี่เริ่มต้นของสัญลักษณ์ในไฟล์ประเภทข้อความ โดยการทดสอบความสำคัญของค่าความถี่เริ่มต้นทีละสัญลักษณ์ด้วยการกำหนดให้มีค่าเป็น 1 ส่วนสัญลักษณ์ที่เหลือมีค่าความถี่คงเดิม จำนวนทั้งหมด 24 สัญลักษณ์ (ตารางที่ 4.4.1 และ ตารางที่ 4.4.2) ทำการตรวจสอบการลดลงของค่าอัตราส่วนการบีบอัด โดยแสดงความสำคัญตลอดจนค่าความน่าจะเป็นเริ่มต้นที่นำเสนอของแต่ละสัญลักษณ์ไว้ดังตารางที่ 4.15

4.3 ผลการทดลอง

จากการออกแบบการทดลองในหัวข้อ 4.2 ทำให้ได้ผลการทดลองแสดงออกมาทั้งในรูปแบบของรูปและตารางดังต่อไปนี้

ตารางที่ 4.4.1 กลุ่มของความน่าจะเป็นบางกลุ่มที่ได้จากการจัดกลุ่มโดยการใช้เทคนิคของการจัดกลุ่มแบบใช้ความสัมพันธ์ตามลำดับชั้น (Gap-7) ของไฟล์จากคลังข้อมูลแคลการีจำนวน 13 ไฟล์ เรียงลำดับความน่าจะเป็นจากค่ามากไปหาค่าน้อย

ลำดับที่	รหัสแอสกี	สัญลักษณ์	ความน่าจะเป็น	กลุ่มที่
1	32	space	0.1371	2
2	101	e	0.0803	3
3	116	t	0.0569	4
4	97	a	0.0468	5
5	105	i	0.0468	5
6	110	n	0.0468	5
7	111	o	0.0468	5
8	115	s	0.0435	6
9	114	r	0.0401	7
10	104	h	0.0334	8
11	10	new line	0.0268	9
12	100	d	0.0268	9
13	108	l	0.0268	9
14	99	c	0.0201	10
15	117	u	0.0201	10
16	44	,	0.0134	11
17	46	.	0.0134	11
18	98	b	0.0134	11
19	102	f	0.0134	11
20	103	g	0.0134	11
21	109	m	0.0134	11
22	112	p	0.0134	11
23	119	w	0.0134	11
24	121	y	0.0134	11
รวม			0.8194	

หมายเหตุ สัญลักษณ์แอสกีที่ไม่ได้นำมาแสดงในตารางนี้อยู่ในกลุ่มที่ 1 และมีค่าความน่าจะเป็นรวมเท่ากับ 18.06

ตารางที่ 4.4.2 ความน่าจะเป็นของสัญลักษณ์บางสัญลักษณ์ที่ได้จากการจัดกลุ่มโดยการใช้เทคนิคของการจัดกลุ่มแบบแบบการแบ่งแยกโดยใช้ค่าเฉลี่ยเค (*k*=14, Gap=6) ของไฟล์จากคลังข้อมูลแคลการีจำนวน 13 ไฟล์ เรียงลำดับความน่าจะเป็นค่าจากมากไปหาค่าน้อย

ที่	รหัสแอสกี	สัญลักษณ์	ความน่าจะเป็น
1	32	space	0.1440
2	101	e	0.0823
3	116	t	0.0578
4	97	a	0.0501
5	111	o	0.0501
6	105	i	0.0476
7	110	n	0.0476
8	115	s	0.0437
9	114	r	0.0411
10	104	h	0.0334
11	10	new line	0.0257
12	100	d	0.0257
13	108	l	0.0257
14	99	c	0.0206
15	109	m	0.0206
16	117	u	0.0206
17	46	.	0.0141
18	102	f	0.0141
19	103	g	0.0141
20	112	p	0.0141
21	44	,	0.0116
22	98	b	0.0116
23	119	w	0.0116
24	121	y	0.0116
	รวม		0.8393

หมายเหตุ สัญลักษณ์แอสกีที่ไม่ได้นำมาแสดงในตารางนี้คือสัญลักษณ์ที่มีค่าความน่าจะเป็นน้อยกว่า 1 % และมีค่าความน่าจะเป็นรวมเท่ากับ 16.07

ตารางที่ 4.4.3 ความน่าจะเป็นของสัญลักษณ์บางสัญลักษณ์ที่ได้จากการใช้การกระจายแบบไวบูลล์ ($\alpha = 0.3, \beta = 6.4$) ของไฟล์นิทาน 9 ไฟล์ เรียงลำดับความน่าจะเป็นจากค่ามากไปหาค่าน้อย

ที่	รหัสแอสกี	สัญลักษณ์	ความน่าจะเป็น
1	32	space	0.1609
2	101	e	0.0891
3	104	h	0.0647
4	116	t	0.0647
5	97	a	0.0517
6	111	o	0.0489
7	110	n	0.0431
8	115	s	0.0417
9	105	i	0.0316
10	114	r	0.0287
11	108	l	0.0259
12	100	d	0.0259
13	117	u	0.0216
14	102	f	0.0187
15	44	,	0.0144
16	109	m	0.0144
17	119	w	0.0101
	รวม		0.7557

หมายเหตุ สัญลักษณ์แอสกีที่ไม่ได้นำมาแสดงในตารางนี้คือสัญลักษณ์ที่มีค่าความน่าจะเป็นน้อยกว่า 1 % และมีค่าความน่าจะเป็นรวมเท่ากับ 24.43

ตารางที่ 4.5 ผลการปรับปรุงค่าอัตราส่วนการบีบอัดของคลังข้อมูลแคลการี โดยการใช้เทคนิคของการจัดกลุ่มแบบการใช้ความสัมพันธ์ตามลำดับชั้นมาใช้ในการเตรียมค่าความถี่เริ่มต้น 1/2 ($\tau = 0.001$, $\varphi = 0.04$, $\delta^* = 15$)

File / Method	Original size (bytes)	1st line: Compression ratio (CR) 2nd line: % increment from CR of IAAC1							
		IAAC1	**Gap-1**	**Gap-2**	**Gap-3**	**Gap-4**	**Gap-5**	**Gap-6**	**Gap-7**
bib	111,262	1.5324	1.5314 (0.0647)	1.5337 0.0855	1.5343 0.1255	1.5346 0.1421	1.5347 0.1490	1.5347 0.1517	1.5348 **0.1545**
book1	768,771	1.7657	1.7658 **0.0078**	1.7651 (0.0317)	1.7649 (0.0425)	1.7642 (0.0819)	1.7648 (0.0494)	1.7645 (0.0675)	1.7644 (0.0730)
book2	610,856	1.6677	1.6779 0.6098	1.6779 0.6092	1.6780 0.6200	1.6793 **0.6936**	1.6776 0.5954	1.6787 0.6598	1.6781 0.6219
news	377,109	1.5396	1.5423 0.1750	1.5488 0.6013	1.5496 0.6505	1.5501 0.6832	1.5506 **0.7196**	1.5498 0.6621	1.5492 0.6252
paper1	53,161	1.5937	1.6233 1.8535	1.6276 2.1217	1.6281 2.1561	1.6284 2.1748	1.6297 **2.2531**	1.6287 2.1905	1.6273 2.1060
paper2	82,199	1.7288	1.7375 0.4988	1.7398 **0.6371**	1.7397 0.6286	1.7395 0.6179	1.7374 0.4946	1.7367 0.4542	1.7368 0.4606
paper3	46,526	1.6995	1.7105 0.6471	1.7103 0.6360	1.7111 0.6841	1.7096 0.5916	1.7104 **0.6397**	1.7093 0.5731	1.7100 0.6175
paper4	13,286	1.6603	1.6983 **2.2881**	1.6946 2.0663	1.6895 1.7548	1.6884 1.6902	1.6882 1.6773	1.6871 1.6127	1.6869 1.5998
paper5	11,954	1.5804	1.6100 1.8721	1.6182 2.3961	1.6253 2.8416	1.6165 2.2853	1.6264 **2.9116**	1.6240 2.7578	1.6180 2.3822
paper6	38,105	1.5816	1.6191 2.3667	1.6298 3.0453	1.6365 3.4702	1.6408 3.7375	1.6371 3.5058	1.6360 3.4391	1.6424 **3.8404**

ตารางที่ 4.5 ผลการปรับปรุงค่าอัตราส่วนการบีบอัดของคลังข้อมูลแคลการี โดยการใช้เทคนิคของการจัดกลุ่มแบบการใช้ความสัมพันธ์ตามลำดับชั้นมาใช้ในการเตรียมค่าความถี่เริ่มต้น 2/2 (τ= 0.001, φ = 0.04, $\delta^* = 15$)

File / Method	Original size (bytes)	1st line: Compression ratio (CR) 2nd line: % increment from CR of IAAC1							
		IAAC1	**Gap-1**	**Gap-2**	**Gap-3**	**Gap-4**	**Gap-5**	**Gap-6**	**Gap-7**
progc	39,611	1.5252	1.5179	1.5311	1.5368	1.5404	1.5442	1.5397	1.5394
			(0.4790)	0.3865	0.7604	0.9955	**1.2436**	0.9484	0.9288
progl	71,646	1.6670	1.6692	1.6816	1.6844	1.6838	1.6851	1.6890	1.6867
			0.1328	0.8754	1.0486	1.0129	1.0866	**1.3225**	1.1818
progp	49,379	1.6299	1.6336	1.6417	1.6412	1.6414	1.6404	1.6446	1.6424
			0.2250	0.7215	0.6913	0.7014	0.6445	**0.8993**	0.7650

หมายเหตุ ตัวเลขที่อยู่ในวงเล็บหมายถึง ค่า CR increment ที่ติดลบ หรือมีค่า CR ที่ไม่ดีขึ้นเมื่อเทียบกับ CR ของ AAC1 และตัวเลขที่เป็นตัวหนาคือค่า CR Increment ที่ดีที่สุดในแต่ละไฟล์

ตารางที่ 4.6 สรุปผลการปรับปรุงค่าอัตราส่วนการบีบอัดของคลังข้อมูลแคลการี โดยการใช้เทคนิคของการจัดกลุ่มแบบการใช้ความสัมพันธ์ตามลำดับชั้นมาใช้ในการเตรียมค่าความถี่เริ่มต้น (τ = 0.001, φ = 0.04, $\delta *= 15$)

	Original size (bytes)	Compression ratio (CR)							
		IAAC1	**Gap-1**	**Gap-2**	**Gap-3**	**Gap-4**	**Gap-5**	**Gap-6**	**Gap-7**
Average CR		1.6286	1.6413	1.6462	1.6477	1.6475	**1.6482**	1.6479	1.6474
Total file size	2,273,865	1,366,424	1,361,895	1,359,910	1,359,486	**1,359,262**	1,359,333	1,359,304	1,359,606
Total Average CR		1.6641	1.6696	1.6721	1.6726	**1.6729**	1.6728	1.6728	1.6724
% Total increment of CR from AAC1		0.000	0.3326	0.4790	0.5103	**0.5269**	0.5217	0.5238	0.5015

หมายเหตุ ตัวเลขที่เป็นตัวหนาคือค่าประจำแต่ละแถวที่มีค่าที่ดีที่สุด

ตารางที่ 4.7 ผลการปรับปรุงค่าอัตราส่วนการบีบอัดของคลังข้อมูลแคลการี โดยการใช้เทคนิคของการจัดกลุ่มแบบการแบ่งแยกโดยใช้ค่าเฉลี่ยเคมาใช้ในการเตรียมค่าความถี่เริ่มต้น 1/2 ($\tau = 0.001$, $\varphi = 0.04$, $\delta^* = 15$, $k = 14$)

File / Method	Original size (bytes)	1st line: Compression ratio (CR) 2nd line: % increment from CR of IAAC1							
		IAAC1	**Gap-1**	**Gap-2**	**Gap-3**	**Gap-4**	**Gap-5**	**Gap-6**	**Gap-7**
bib	111,262	1.5324	1.5267 (0.3746)	1.5314 (0.0674)	1.5328 0.0276	1.5335 0.0717	1.5338 0.0896	1.5343 0.1227	1.5344 **0.1269**
book1	768,771	1.7657	1.7604 (0.2965)	1.7637 **(0.1133)**	1.7632 (0.1365)	1.7636 (0.1142)	1.7635 (0.1213)	1.7635 (0.1195)	1.7635 (0.1218)
book2	610,856	1.6677	1.6771 0.5620	1.6781 0.6266	1.6803 0.7540	1.6779 0.6101	1.6797 0.7221	1.6804 **0.7617**	1.6780 0.6156
news	377,109	1.5396	1.5359 (0.2374)	1.5461 0.4231	1.5475 0.5129	1.5489 0.6033	1.5508 **0.7304**	1.5503 0.6956	1.5492 0.6244
paper1	53,161	1.5937	1.6173 1.4786	1.6168 1.4446	1.6260 2.0217	1.6190 1.5836	1.6295 **2.2406**	1.6276 2.1248	1.6286 2.1842
paper2	82,199	1.7288	1.7369 0.4691	1.7383 0.5456	1.7377 0.5137	1.7374 0.4967	1.7374 0.4946	1.7379 0.5222	1.7390 **0.5881**
paper3	46,526	1.6995	1.7082 0.5103	1.7104 0.6397	1.7111 0.6804	1.7111 0.6804	1.7111 0.6804	1.7115 **0.7026**	1.7096 0.5953
paper4	13,286	1.6603	1.6869 1.5998	1.6951 **2.0924**	1.6895 1.7548	1.6893 1.7419	1.6895 1.7548	1.6895 1.7548	1.6888 1.7160
paper5	11,954	1.5804	1.6054 1.5847	1.6124 2.0232	1.6196 2.4793	1.6165 2.2853	1.6218 2.6184	1.6196 2.4793	1.6251 **2.8276**
paper6	38,105	1.5816	1.6124 1.9422	1.6273 2.8869	1.6339 3.3060	1.6325 3. 2175	1.6349 3.3681	1.6417 **3.7956**	1.6363 3.4524

ตารางที่ 4.7 ผลการปรับปรุงค่าอัตราส่วนการบีบอัดของคลังข้อมูลแคลการี โดยการใช้เทคนิคของการจัดกลุ่มแบบการแบ่งแยกโดยใช้ค่าเฉลี่ยเคมาใช้ในการเตรียมค่าความถี่เริ่มต้น 2/2 (τ = 0.001, φ = 0.04, $\delta^* = 15$, k = 14)

File / Method	Original size (bytes)	1st line: Compression ratio (CR) 2nd line: % increment from CR of IAAC1							
		IAAC1	**Gap-1**	**Gap-2**	**Gap-3**	**Gap-4**	**Gap-5**	**Gap-6**	**Gap-7**
progc	39,611	1.5252	1.4996 (1.6771)	1.5286 0.2200	1.5312 0.3943	1.5352 0.6550	1.5422 1.1174	1.5406 1.0073	1.5434 **1.1923**
progl	71,646	1.6670	1.6652 (0.1046)	1.6727 0.3455	1.6779 0.6581	1.6801 0.7903	1.6830 0.9631	1.6849 **1.0771**	1.6836 0.9963
progp	49,379	1.6299	1.6258 (0.2535)	1.6371 0.4376	1.6400 0.6178	1.6413 0.6980	1.6453 **0.9396**	1.6431 0.8086	1.6435 0.8321

หมายเหตุ ตัวเลขที่อยู่ในวงเล็บหมายถึง ค่า CR ที่ติดลบ หรือมีค่า CR ที่ไม่ดีขึ้นเมื่อเทียบกับ CR ของ IAAC1 และตัวเลขที่เป็นตัวหนาคือค่า CR Increment ที่ดีที่สุดในแต่ละไฟล์

ตารางที่ 4.8 สรุปผลการปรับปรุงค่าอัตราส่วนการบีบอัดของคลังข้อมูลแคลการี โดยการใช้เทคนิคของการจัดกลุ่มแบบการใช้ความสัมพันธ์ตามลำดับชั้นมาใช้ในการเตรียมค่าความถี่เริ่มต้น (τ = 0.001, φ = 0.04, $\delta^* = 15$, k = 14)

	Original size (bytes)	**IAAC1**	**Gap-1**	**Gap-2**	**Gap-3**	**Gap-4**	**Gap-5**	**Gap-6**	**Gap-7**
Average CR		1.6286	1.6352	1.6429	1.6454	1.6451	1.6479	**1.6481**	1.6479
Total file size	2,273,865	1,366,424	1,365,543	1,361,420	1,360,258	1,360,467	1,359,241	**1,359,101**	1,359,822
Total Average CR		1.6641	1.6652	1.6702	1.6716	1.6714	1.6729	**1.6731**	1.6722
% Total increment of CR from IAAC1		0.0000	0.0645	0.3676	0.4533	0.4379	0.5285	**0.5388**	0.4855

หมายเหตุ ตัวเลขที่เป็นตัวหนาคือค่าประจำแต่ละแถวที่มีค่าที่ดีที่สุด

ตารางที่ 4.9 ผลการหาค่าพารามิเตอร์แสดงรูปร่าง และพารามิเตอร์มาตราส่วนที่ให้ค่าอัตราส่วนการบีบอัดของกลุ่มของไฟล์นิทานภาษาอังกฤษ($\beta = 6.4$)

File / α	**Original size (bytes)**	**1st line: Compression ratio (CR)** **2nd line: % increment from CR of IAAC1**					
		IAAC1	**0.1**	**0.3**	**0.5**	**0.7**	**0.9**
A Tale of Two Cities	776,629	1.7830	1.7852 0.1240	1.7855 **0.1403**	1.7843 0.0738	1.7830 0.0014	1.7793 (0.2064)
Ethan Frome	203,305	1.7835	1.7883 0.2684	1.7928 **0.5222**	1.7850 0.0818	1.7748 (0.4870)	1.7689 (0.8169)
Heart of Darkness	229,831	1.7764	1.7814 0.2807	1.7839 **0.4208**	1.7808 0.2473	1.7736 (0.1588)	1.7668 (0.5426)
Moby Dick	1,231,973	1.7822	1.7846 0.1369	1.7853 **0.1713**	1.7841 0.1087	1.7815 (0.0387)	1.7816 (0.0322)
Native Son	32,020	1.6816	1.6854 0.2284	1.6989 **1.0260**	1.6863 0.2812	1.6729 (0.5152)	1.6549 (1.5898)
Pride and Prejudice	704,158	1.7859	1.7874 0.0844	1.7884 **0.1408**	1.7870 0.0633	1.7843 (0.0906)	1.7818 (0.2322)
Robinson Crusoe	642,573	1.8426	1.8429 0.0174	1.8444 **0.1001**	1.8418 (0.0411)	1.8369 (0.3120)	1.8337 (0.4832)
Silas Marner	413,529	1.7865	1.7880 0.0862	1.7909 **0.2453**	1.7882 0.0966	1.7837 (0.1564)	1.7745 (0.6731)
The Invisible Man	292,663	1.7505	1.7518 0.0726	1.7559 **0.3109**	1.7486 (0.1092)	1.7475 (0.1706)	1.7306 (1.1358)

หมายเหตุ ตัวเลขที่อยู่ในวงเล็บหมายถึง ค่า CR increment ที่ติดลบ หรือมีค่า CR ที่ไม่ดีขึ้นเมื่อเทียบกับ CR ของ IAAC1 และตัวเลขที่เป็นตัวหนาคือค่า CR Increment ที่ดีที่สุดในแต่ละไฟล์

ตารางที่ 4.10 สรุปผลการหาค่าพารามิเตอร์แสดงรูปร่างและ พารามิเตอร์มาตราส่วนที่ให้ค่าอัตราส่วนการบีบอัดของกลุ่มของไฟล์นิทานภาษาอังกฤษ ($\beta = 6.4$)

Issue / α	**file size (bytes)**	**IAAC1**	**0.1**	**0.3**	**0.5**	**0.7**	**0.9**
Average CR	N.A.	0.0000	1.7747	1.7772	**1.7807**	1.7762	1.7709
% increment of CR from IAAC1			0.1443	**0.3420**	0.0892	(0.2142)	(0.6347)
Total file size	4,526,681	2,530,941	2,528,026	**2,525,827**	2,529,251	2,534,156	2,540,111
Total Average CR	N.A.	1.7885	1.7906	**1.7922**	1.7897	1.7863	1.7821
% Total increment of CR from IAAC1	N.A.	0.0000	0.1153	**0.2025**	0.0668	(0.1269)	(0.3610)

หมายเหตุ ตัวเลขที่อยู่ในวงเล็บหมายถึง ค่า CR increment ที่ติดลบ หรือมีค่า CR ที่ไม่ดีขึ้นเมื่อเทียบกับ CR ของ IAAC1 และตัวเลขที่เป็นตัวหนาคือค่าประจำแต่ละแถวที่มีค่าที่ดีที่สุด

ตารางที่ 4.11 ผลการทวนสอบประสิทธิผลของอัลกอริทึมในการปรับปรุงการเข้ารหัสเลขคณิตส่วนเพิ่มขึ้นที่สามารถปรับเปลี่ยนค่าความน่าจะเป็นได้ กับคลังข้อมูลแคนเทอเบอร์รีขนาดใหญ่

File / method	1st line: Compression ratio (CR) 2nd line: % increment from CR of IAAC1				
	Original file size (bytes)	**IAAC1**	**I**	**II**	**III**
bible.txt	4,047,392	1.8418	1.8424	**1.8432**	1.8424
		0.0000	0.0363	**0.0784**	0.0350
e.coli	4,638,690	3.9986	**3.4842**	3.4815	3.4821
		0.0000	**-12.8649**	-12.9334	-12.9173
world.txt	2,473,400	1.6001	1.6011	**1.6032**	1.6029
		0.0000	0.0621	**0.1925**	0.1737

หมายเหตุ I. Hierarchical clustering ($\tau = 0.001$, $\varphi = 0.04$, $\delta^* = 15$, Gap-4)
II. K-means clustering ($\tau = 0.001$, $\varphi = 0.04$, $\delta^* = 15$, $k = 14$, Gap-6)
III. Weibull dist. ($\tau = 0.001$, $\varphi = 0.04$, $\delta^* = 15$, $\alpha = 0.3, \beta = 6.4$)
ตัวหนา คือค่า CR increment ที่ดีที่สุดในแต่ละไฟล์

ตารางที่ 4.12 สรุปผลการทวนสอบประสิทธิผลของอัลกอริทึมในการปรับปรุงการเข้ารหัสเลขคณิตส่วนเพิ่มขึ้นที่สามารถปรับเปลี่ยนค่าความน่าจะเป็นได้ กับคลังข้อมูลแคนเทอเบอร์รีขนาดใหญ่

Issue / α	file size (bytes)	IAAC1	I	II	III
Total file size	11,159,482	4,903,365	4,903,163	**4,901,142**	4,902,171
Total Average CR	N.A.	2.2759	2.2760	**2.2769**	2.2764
% Total increment of CR from IAAC1	N.A.	0.0000	0.0041	**0.0453**	0.0243

หมายเหตุ I. Hierarchical clustering ($\tau = 0.001$, $\varphi = 0.04$, $\delta^* = 15$, Gap-4)
II ($\tau = 0.001$, $\varphi = 0.04$, $\delta^* = 15$, $k = 14$, Gap-6)
III. Weibull dist. ($\tau = 0.001$, $\varphi = 0.04$, $\delta^* = 15$, $\alpha = 0.3, \beta = 6.4$)
ตัวเลขที่เป็นตัวหนาคือค่าประจำแต่ละแถวที่มีค่าที่ดีที่สุด

ตารางที่ 4.13 ผลการทวนสอบประสิทธิผลของอัลกอริทึมในการปรับปรุงการเข้ารหัสเลขคณิตส่วนเพิ่มขึ้นที่สามารถปรับเปลี่ยนค่าความน่าจะเป็นได้ กับไฟล์นิทานภาษาอังกฤษ

File / Method	Original size (bytes)	1st line: Compression ratio (CR) 2nd line: % increment from CR of IAAC1			
		IAAC1	**I**	**II**	**III**
A Tale of Two Cities	776,629	1.7830	1.7855 0.1403	1.7858 0.1578	1.7861 **0.1714**
Ethan Frome	203,305	1.7835	1.7928 **0.5222**	1.7900 0.3629	1.7905 0.3947
Heart of Darkness	229,831	1.7764	1.7839 0.4208	1.7838 0.4169	1.7844 **0.4489**
Moby Dick	1,231,973	1.7822	1.7853 0.1713	1.7859 **0.2078**	1.7855 0.1829
Native Son	32,020	1.6816	1.6989 1.0260	1.7032 **1.2840**	1.7015 1.1817
Pride and Prejudice	704,158	1.7859	1.7884 0.1408	1.7886 0.1497	1.7889 **0.1665**
Robinson Crusoe	642,573	1.8426	1.8444 0.1001	1.8449 0.1262	1.8450 **0.1308**
Silas Marner	413,529	1.7865	1.7909 0.2453	1.7913 0.2683	1.7914 **0.2770**
The Invisible Man	292,663	1.7505	1.7559 0.3109	1.7557 0.2947	1.7563 **0.3302**

หมายเหตุ I. Weibull dist. ($\tau = 0.001$, $\varphi = 0.04$, $\delta^* = 15$, $\alpha = 0.3$, $\beta = 6.4$)

II. Hierarchical clustering ($\tau = 0.001$, $\varphi = 0.04$, $\delta^* = 15$, Gap-4)

III. K-means clustering ($\tau = 0.001$, $\varphi = 0.04$, $\delta^* = 15$, $k = 14$, Gap-6)

ตัวเลขที่เป็นตัวหนาคือค่า CR Increment ที่ดีที่สุดในแต่ละไฟล์

ตารางที่ 4.14 สรุปผลการทวนสอบประสิทธิผลของอัลกอริทึมในการปรับปรุงการเข้ารหัสเลขคณิตส่วนเพิ่มขึ้นที่สามารถปรับเปลี่ยนค่าความน่าจะเป็นได้ กับไฟล์นิทานภาษาอังกฤษ

Issue / **Method**	**file size (bytes)**	**IAAC1**	**I**	**II**	**III**
Average CR % increment of CR from IAAC1	N.A.	0.0000	1.7772 0.3420	1.7810 0.3565	1.7811 **0.3589**
Total file size	4,526,681	2,530,941	2,525,827	2,525,485	**2,525,378**
Total Average CR	N.A.	1.7885	1.7922	1.7924	**1.7925**
% Total increment of CR from IAAC1	N.A.	0.0000	0.2025	0.2181	**0.2223**

หมายเหตุ I. Weibull dist. ($\tau = 0.001$, $\varphi = 0.04$, $\delta^* = 15$, $\alpha = 0.3$, $\beta = 6.4$)
II. Hierarchical clustering ($\tau = 0.001$, $\varphi = 0.04$, $\delta^* = 15$, Gap-4)
III. K-means clustering ($\tau = 0.001$, $\varphi = 0.04$, $\delta^* = 15$, $k = 14$, Gap-6)
ตัวเลขที่เป็นตัวหนาคือค่าประจำแต่ละแถวที่มีค่าที่ดีที่สุด

ตารางที่ 4.15 สรุปความสำคัญของสัญลักษณ์ต่อค่าความถี่เริ่มต้นสำหรับการเข้ารหัสเลขคณิตส่วนเพิ่มขึ้นที่สามารถปรับเปลี่ยนค่าความน่าจะเป็นได้

รหัสแอสกี	**สัญลักษณ์**	**ความสำคัญ (**t-value, df, sig. (2-tailed))	
		I	II
32	space	☑ (4.968, 12, 0)	☑ (6.103096, 12, 0)
101	e	☐ (1.530234, 12, 0.152)	☑ (3.52468, 12, 0.004)
116	t	☐ (1.705129, 12, 0.114)	☐ (0.089743,12, 0.93)
97	a	☐ (-0.46407, 12, 0.651)	☐ (1.306091, 12, 0,216)
115	s	☐ (-0.18715,12, 0.855)	☐ (0.688436, 12, 0.504)

หมายเหตุ I. Hierarchical clustering ($\tau = 0.001$, $\varphi = 0.04$, $\delta^* = 15$, Gap-4)
II. K-means clustering ($\tau = 0.001$, $\varphi = 0.04$, $\delta^* = 15$, $k = 14$, Gap-6)

4.4 ผลการวิเคราะห์และเปรียบเทียบ

4.4.1 ผลการวิเคราะห์

จากการออกแบบการทดลองทั้งสี่ ในหัวข้อ 4.2.1 - 4.2.4 ดังกล่าวข้างต้น ทุกการทดลอง (ตารางที่ 4.5 – 4.12) สามารถให้ค่าอัตราส่วนการบีบอัดในแนวโน้มที่ดีขึ้น เมื่อเปรียบเทียบกับการใช้แต่อัลกอริทึม IAAC1 ทั้งในคลังข้อมูลแคลการีจำนวน 13 ไฟล์ คลังข้อมูลแคนเทอเบอร์รีขนาดใหญ่จำนวน 3 ไฟล์ และกลุ่มของไฟล์นิทานภาษาอังกฤษจำนวน 9 ไฟล์ โดยผลการเปรียบเทียบและผลการวิเคราะห์ปัจจัยในมุมมองต่างๆ มีดังนี้

4.4.2 ผลการเปรียบเทียบและผลการวิเคราะห์ปัจจัย

4.4.2.1 ผลการวิเคราะห์ปัจจัยด้านค่าความถี่เริ่มต้น

จากตารางที่ 4.4.1 และ 4.4.2 นั้น ในแต่ละตารางมีสัญลักษณ์แอสกีจำนวนเท่ากัน อีกทั้งมีสัญลักษณ์ที่เป็นสมาชิกในกลุ่มที่มีค่าความน่าจะเป็นสูงกว่า 0.01 เหมือนกัน (ลำดับที่ 1 – 14) นอกจากนั้นการใช้การแบ่งแยกแบบใช้ค่าเฉลี่ยเคและการใช้ความสัมพันธ์ตามลำดับชั้น การใช้การแบ่งแยกแบบใช้ค่าเฉลี่ยเคนั้นจะให้ค่าความน่าจะเป็นของสัญลักษณ์หลายตัว เช่นในลำดับที่ 1-10 มีค่าสูงกว่าการจัดกลุ่มแบบการใช้ความสัมพันธ์ตามลำดับชั้น โดยพิจารณาจากการคำนวนทางสถิติดังในตารางที่ 4.16 ดังนั้นสมมติฐานโดยเบื้องต้นจะได้ว่า การเข้ารหัสเลขคณิตส่วนเพิ่มขึ้นที่สามารถปรับเปลี่ยนค่าความน่าจะเป็นได้ โดยการใช้การแบ่งแยกแบบใช้ค่าเฉลี่ยเค จะสามารถให้ค่าอัตราส่วนการบีบอัดที่ดีกว่าการใช้ค่าความสัมพันธ์ตามลำดับชั้นมาใช้ในการเตรียมค่าความถี่เริ่มต้น

ในตารางที่ 4.4.3 นั้น ใช้ไฟล์ bible.txt เป็นฐานในการสร้างค่าความถี่เริ่มต้นโดยใช้การกระจายแบบไวบูลล์ ดังนั้นจำนวนและสมาชิกสัญลักษณ์ที่ได้ จึงมีความแตกต่างจากตารางที่ 4.4.1 และ 4.4.2 โดยมีสัญลักษณ์จำนวน 7 สัญลักษณ์ที่ขาดหายไปในตารางที่ 4.4.3 ได้แก่ ‘newline’, ‘c’, ‘.’, ‘b’, ‘g’, ‘p’ และ ‘y’ ซึ่งเป็นปัจจัยสำคัญปัจจัยหนึ่งต่อการเพิ่มค่าอัตราส่วนการบีบอัด

ตารางที่ 4.16 ผลการเปรียบเทียบความแตกต่างของค่าอัตราส่วนการบีบอัดระหว่างการใช้การแบ่งแยกแบบใช้ค่าเฉลี่ยเค และการใช้ค่าความสัมพันธ์ตามลำดับชั้นมาใช้ในการเตรียมค่าความถี่เริ่มต้น มาใช้ในการเข้ารหัสเลขคณิตส่วนเพิ่มขึ้นที่สามารถปรับเปลี่ยนค่าความน่าจะเป็นได้ ของคลังข้อมูลแคลการี

Paired Samples Test

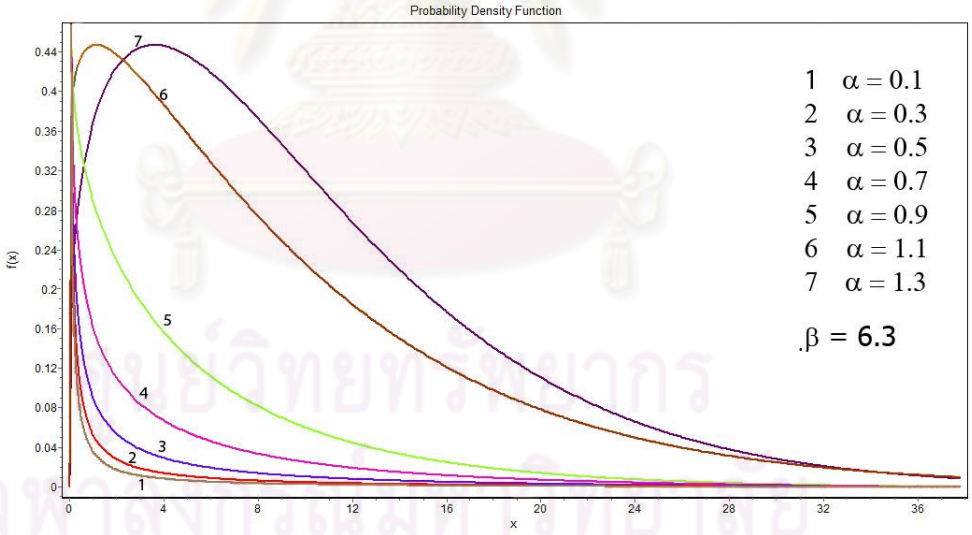
	Paired Differences					t	df	Sig. (2-tailed)
				95% Confidence Interval of the Difference				
	Mean	Std. Deviation	Std. Error Mean	Lower	Upper			
Pair 1 h - k	-5.076E-4	.0031933	.0008857	-.0024374	.0014220	-.573	12	.577

4.4.2.2 ผลการวิเคราะห์ปัจจัยด้านการปรับปรุงค่าอัตราส่วนการบีบอัด

จากตารางที่ 4.5 เป็นผลการปรับปรุงค่าอัตราส่วนการบีบอัดของคลังข้อมูลแคลการี โดยการใช้เทคนิคของการจัดกลุ่มแบบการใช้ความสัมพันธ์ตามลำดับชั้นมาใช้ในการเตรียมค่าความถี่เริ่มต้น (τ = 0.001, φ = 0.04, $\delta^* = 15$) ค่าอัตราส่วนการบีบอัดที่เพิ่มขึ้นจะมีค่าสูงสุดที่การใช้ค่า $\mathbb{N} = 7$ ที่ไฟล์ paper6 ซึ่งมีค่าเท่ากับ 3.8404% และมีค่าต่ำสุดที่ $\mathbb{N} = 7$ เช่นเดียวกัน ที่ไฟล์ book1 ซึ่งมีค่าเท่ากับ -0.0730% แต่เมื่อเปรียบเทียบกับผลในตารางที่ 4.7 การใช้เทคนิคของการจัดกลุ่มแบบการแบ่งแยกโดยใช้ค่าเฉลี่ยเคมาใช้ในการเตรียมค่าความถี่เริ่มต้นแล้ว (τ = 0.001, φ = 0.04, $\delta^* = 15$, k = 14) จะให้ผลลัพธ์ที่แตกต่างกันออกไปเล็กน้อย โดยค่าอัตราส่วนการบีบอัดที่เพิ่มขึ้นจะมีค่าสูงสุดที่การใช้ค่า $\mathbb{N} = 6$ ที่ไฟล์ paper6 ซึ่งมีค่าเท่ากับ 3.7956% และมีค่าต่ำสุดที่ ค่า $\mathbb{N} = 1$ ที่ไฟล์ bib ซึ่งมีค่าเท่ากับ -0.3746% และเมื่อทำการทดลองให้ค่า $\mathbb{N}$ มีค่ามากกว่า 7 แล้ว พบว่า ค่าอัตราส่วนการบีบอัดที่ได้จากทั้งการใช้การใช้เทคนิคของการจัดกลุ่มแบบการแบ่งแยกโดยใช้ค่าเฉลี่ยเค และการใช้ความสัมพันธ์ตามลำดับชั้น ให้ค่าอัตราส่วนการบีบอัดที่น้อยกว่า

จากตารางที่ 4.9 เป็นผลการหาค่าพารามิเตอร์แสดงรูปร่างและพารามิเตอร์มาตราส่วนของการกระจายแบบไวบูลล์ ที่ส่งผลให้ค่าอัตราส่วนการบีบอัดของกลุ่มของไฟล์นิทานภาษาอังกฤษ ซึ่งมีค่าเท่ากับ 0.3 และ 6.4 ตามลำดับ โดยมีค่าอัตราส่วนการบีบอัดที่เพิ่มขึ้นสูงสุดเท่ากับ 1.0260% ที่ไฟล์ Native Son และมีค่าอัตราส่วนการบีบอัดที่เพิ่มขึ้นต่ำสุดเท่ากับ -1.5898% ที่ค่า α = 0.9 ที่ไฟล์ Native Son เช่นกัน หากพิจารณาการเพิ่มขึ้นของค่า α ค่าอัตราส่วนการบีบอัดที่เพิ่มขึ้นจะมีค่าลดลงเนื่องจากสาเหตุคือ ค่าความถี่หรือค่าความถี่เริ่มต้นของสัญลักษณ์ที่มีความถี่สูงมีค่าใกล้เคียงกัน อีกทั้งจะใกล้เคียงกันมากขึ้นเมื่อค่า α มีค่าเพิ่มขึ้น ซึ่งจะส่งผลให้ปรากฏในกราฟของฟังก์ชันความหนาแน่นของความน่าจะเป็นดังรูปที่ 4.1

จากรูปที่ 4.1พบว่าเมื่อค่า α มีค่าเพิ่มขึ้นแล้ว รูปร่างของกราฟฟังก์ชันความหนาแน่นของความน่าจะเป็นของการกระจายแบบไวบูลล์ ได้เปลี่ยนรูปร่างให้มีลักษณะเหมือนกับเส้นกราฟฟังก์ชันความหนาแน่นของความน่าจะเป็นของการกระจายแบบปกติ (normal distribution) ดังเส้นกราฟที่ 7 ($\alpha = 1.3$) แต่ในความเป็นจริงแล้ว สัญลักษณ์ในไฟล์ข้อความภาษาอังกฤษ ที่ยึดเอาไฟล์ bible.txt เป็นต้นแบบนั้น จะมีลักษณะใกล้เคียงกับเส้นกราฟที่ 1, 2 และ 3 มากกว่า ($\alpha = 0.1, 0.3, 0.5$) ตามลำดับ ซึ่งค่า α และ i ที่ได้จากการ fit distribution ด้วยโปรแกรม input analyzer ของไฟล์นี้นั้น จะมีค่าเท่ากับ 0.125 และ 2.42 ตามลำดับ แต่ในการทดลองนี้ให้เสนอให้ใช้ค่า $\alpha = 0.3$ (กราฟเส้นที่ 2) ซึ่งเส้นกราฟของค่า α ที่นำเสนอนี้มีรูปร่างใกล้เคียงกับเส้นกราฟที่ใช้ค่า $\alpha = 0.1$ (กราฟเส้นที่ 1) ด้วยเหตุผลนี้จึงทำให้สรุปได้ว่า เมื่อใช้การกระจายแบบไวบูลล์ที่มีค่าพารามิเตอร์ด้านรูปร่างที่ค่า $\alpha \leq 0.3$ มาใช้ในการประมาณค่าความถี่เริ่มต้นของการเข้ารหัสเลขคณิตส่วนเพิ่มขึ้น ที่สามารถปรับเปลี่ยนค่าความน่าจะเป็นได้แล้ว นอกจากจะสามารถเลียนแบบพฤติกรรมของการกระจายของสัญลักษณ์ในเอกสารภาษาอังกฤษได้แล้ว ยังสามารถจะก่อให้เกิดค่าอัตราส่วนการบีบอัดที่เพิ่มขึ้นได้อีกด้วย เมื่อเทียบกับ IAAC1



รูปที่ 4.1 ฟังก์ชันความหนาแน่นของความน่าจะเป็นของการกระจายแบบไวบูลล์ที่ค่า α และ β ต่างๆ กัน

ตารางที่ 4.11 เป็นการทวนสอบประสิทธิผลของอัลกอริทึมในการประมาณค่าความถี่เริ่มต้น โดยนำค่าพารามิเตอร์ต่างๆ ที่ก่อให้เกิดผลที่ดีที่สุดในแต่ละอัลกอริทึม มาใช้ในการเข้ารหัสของไฟล์ในคลังข้อมูลแคนเทอเบอร์รีขนาดใหญ่ ซึ่ง

จากผลลัพธ์ในตารางที่ 4.11 นั้น แสดงให้เห็นว่า วิธีที่ II หรือการใช้เทคนิคของการจัดกลุ่มแบบการแบ่งแยกโดยใช้ค่าเฉลี่ยเค ($k = 14$) จะให้ผลลัพธ์ที่ดีที่สุดในการเข้ารหัสเลขคณิตและถอดรหัสเลขคณิตส่วนเพิ่มขึ้น ที่สามารถปรับเปลี่ยนค่าความน่าจะเป็นได้ ส่วนไฟล์ e.coli ให้ค่าการบีบอัดที่ไม่ดีขึ้นเมื่อเทียบกับ IAAC1 เนื่องจากไฟล์นี้ประกอบด้วยสัญลักษณ์เพียงสี่สัญลักษณ์เท่านั้นคือ ‘a’, ‘g’, ‘t’ และ ‘c’ ดังนั้นผลการใช้การประมาณค่าด้วยเทคนิคทั้งสามดังกล่าวข้างต้น จึงทำให้ไม่มีประสิทธิภาพในการเข้ารหัสไฟล์ดังกล่าว ส่วนไฟล์ bible.txt และ world.txt นั้น เมื่อนำมาวิเคราะห์ค่าความน่าจะเป็นของแต่ละสัญลักษณ์เพื่อเปรียบเทียบกับ ค่าความน่าจะเป็นที่คำนวนจากค่าความถี่เริ่มต้นของการใช้เทคนิคของการจัดกลุ่มแบบการแบ่งแยกโดยใช้ค่าเฉลี่ยเค ($k = 14$) นั้นสามารถแสดงได้ดังในตารางที่ 4.17

ตารางที่ 4.17 ผลการเปรียบเทียบค่าความน่าจะเป็นของแต่ละสัญลักษณ์ ในรหัสแอสกีบางตัว ที่มีค่าความน่าจะเป็นในระดับสูง ระหว่างไฟล์ bible.txt world.txt และค่าความน่าจะเป็นที่ได้จากเทคนิคของการจัดกลุ่มแบบการแบ่งแยกโดยใช้ค่าเฉลี่ยเค ($k = 14$)

ลำดับ	สัญลักษณ์	ความน่าจะเป็น		
		bible.txt	world.txt	k-mean (k=14)
1	space	0.1893	0.1733	0.1440
2	e	0.0979	0.0659	0.0823
3	t	0.0740	0.0464	0.0578
4	h	0.0668	0.0152	0.0334
5	a	0.0615	0.0595	0.0501
6	o	0.0559	0.0447	0.0501
7	n	0.0532	0.0483	0.0476
8	s	0.0442	0.0358	0.0437
9	i	0.0430	0.0482	0.0476
10	r	0.0389	0.0452	0.0411
11	d	0.0356	0.0206	0.0257
12	l	0.0290	0.0296	0.0257
13	Newline	0.0075	0.0263	0.0257

จากตารางที่ 4.17 นี้เอง ทำให้พอทราบได้ว่า ค่าความน่าจะเป็นของสัญลักษณ์ ‘space’ ที่ได้จากการใช้เทคนิคของการจัดกลุ่มแบบการแบ่งแยกโดยใช้ค่าเฉลี่ยเค ($k = 14$) มีค่าแตกต่างจากค่าความน่าจะเป็นของสัญลักษณ์นี้ในไฟล์ bible.txt และ world.txt มาก อีกทั้งความน่าจะเป็นของแต่ละสัญลักษณ์ มีการกระจายต่างกันด้วย ซึ่งเป็นเหตุผลหลักที่ทำให้ผลการทดลองในตารางที่ 4.11 ให้ค่าอัตราส่วนการบีบอัดไม่ดี

จากตารางที่ 4.13 และ 4.14 เป็นการยืนยันว่า การใช้ค่าความถี่เริ่มต้นจากไฟลข้อความทั่วไปที่ได้จากไฟล์ในคลังข้อมูลแคลการีด้วยวิธีการใช้ความสัมพันธ์แบบลำดับชั้น สามารถมีค่าอัตราส่วนการบีบอัดที่ดีที่สุด (1.2840) ได้สูงกว่าการใช้การกระจายแบบไวบูลล์ แต่อย่างไรก็ตาม เมื่อพิจารณาในภาพรวมพบว่า การใช้การจัดกลุ่มแบบการแบ่งแยกโดยใช้ค่าเฉลี่ยเค ($k = 14$) นั้นให้จำนวนไฟล์ที่มีค่า CR increment สูงที่สุดในแต่ละไฟล์ ด้วยจำนวน 6 ไฟล์ จาก 13 ไฟล์ ดังนั้นในงานวิจัยครั้งนี้ จึงสรุปให้การจัดกลุ่มแบบการแบ่งแยกโดยใช้ค่าเฉลี่ยเค ($k = 14$) นี้เป็นวิธีที่ดีที่สุดในการเตรียมค่าความถี่เริ่มต้นสำหรับเอกสารภาษาอังกฤษ ที่จะสามารถให้ค่าอัตราส่วนการบีบอัดที่ดีที่สุดได้

ตารางที่ 4.15 สัญลักษณ์ที่มีความสำคัญต่อค่าอัตราส่วนในการบีบอัดของเอกสารประเภทข้อความภาษาอังกฤษคือ ‘space’ และ ตัว ‘e’ โดยเมื่อกำหนดให้ความถี่เริ่มต้นของทั้งสองสัญลักษณ์นี้มีค่าต่ำสุดแล้ว จะทำให้ค่าอัตราส่วนการบีบอัดที่ได้ มีค่าน้อยลงอย่างมีระดับนัยสำคัญ แต่อย่างไรก็ตาม สัญลักษณ์ที่เหลืออีก 22 สัญลักษณ์ต่างก็ทำให้ค่าอัตราส่วนการบีบอัดลดลงได้เช่นกัน

บทที่ 5
สรุปผลการวิจัย อภิปรายผล และข้อเสนอแนะ

เทคนิคต่างๆ ที่ได้นำเสนอในงานวิจัยนี้ มีวัตถุประสงค์เพื่อเพิ่มค่าอัตราส่วนการบีบอัดของเอกสารประเภทข้อความภาษาอังกฤษ ที่ใช้การเข้ารหัสแบบเลขคณิตส่วนเพิ่มขึ้นที่สามารถปรับเปลี่ยนค่าความน่าจะเป็นได้ เทคนิคต่างๆ ที่นำมาใช้คือการจัดกลุ่มแบบการใช้ความสัมพันธ์ตามลำดับชั้น และการแบ่งแยกโดยใช้ค่าเฉลี่ยเคในการเตรียมค่าความน่าจะเป็นเริ่มต้น ตลอดจนการกระชับค่าความความน่าจะเป็นเริ่มต้นด้วย การลดความแตกต่างของค่าความน่าจะเป็นของสัญลักษณ์ที่มีความถี่สูงสุดและความถี่ต่ำสุดให้น้อยลง การลดทอนสัญลักษณ์ที่ไม่ปรากฏ นอกจากนี้ในระหว่างการเข้ารหัส ยังได้เพิ่มการพิจารณาการแบ่งไฟล์ออกเป็นส่วนๆ เมื่อพบว่าค่าอัตราส่วนการบีบอัดในขณะนั้นมีค่าคงที่หรือมีค่าลดลง ซึ่งล้วนแล้วแต่เป็นการเพิ่มค่าอัตราส่วนการบีบอัดของเอกสารข้อความภาษาอังกฤษได้เป็นอย่างดี อีกทั้งการประมาณค่าความน่าจะเป็นเริ่มต้นด้วยการใช้การกระจายแบบไวบูลล์ก็เป็นอีกทางเลือกหนึ่งที่ให้ค่าผลลัพธ์ไปในทิศทางเดียวกัน

การจัดกลุ่มแบบการใช้ความสัมพันธ์ตามลำดับชั้น การแบ่งแยกโดยใช้ค่าเฉลี่ยเค และการใช้การกระจายแบบไวบูลล์ต่างก็เป็นเทคนิคที่ใช้ในการเตรียมค่าความถี่เริ่มต้น ซึ่งเทคนิคดังกล่าวเหล่านั้นจำเป็นที่จะต้องมีการนำข้อมูลทดสอบมาเรียนรู้ไว้ก่อน จึงจะสามารถให้ค่าความถี่เริ่มต้นได้ โดยในการใช้การกระจายแบบไวบูลล์ จะมีความซับซ้อนของอัลกอริทึมการเตรียมค่าความถี่เริ่มต้นที่น้อยกว่า อีกทั้งสามารถให้ค่าอัตราส่วนการบีบอัดที่เพิ่มขึ้นอยู่ในเกณฑ์ดี หากข้อมูลที่จะเข้ารหัส มีค่าความน่าจะเป็นของสัญลักษณ์ หรือค่าการกระจายแตกต่างออกไปจากการกระจายแบบไวบูลล์ งานวิจัยที่นำเสนอมาก็สามารถนำไปประยุกต์ใช้ได้ เพื่อก่อให้เกิดอัตราส่วนการบีบอัดที่เพิ่มขึ้นได้

เมื่อเปรียบเทียบค่าอัตราส่วนการบีบอัดที่ได้จากการทดลองทั้งหมดพบว่า การใช้เทคนิคการเตรียมค่าความถี่เริ่มต้นด้วยการใช้การจัดกลุ่มแบบการแบ่งแยกโดยใช้ค่าเฉลี่ยเค ที่ระดับจำนวนกลุ่มเท่ากับ 14 ($k = 14$) นั้น เป็นค่าความถี่เริ่มต้นที่ดีที่สุด (ตารางที่ 4.4.2) สำหรับการเข้ารหัสด้วยวิธีที่นำเสนอในงานวิจัยนี้ โดยจะสามารถเพิ่มค่าอัตราส่วนการบีบอัดได้เพิ่มขึ้นเมื่อเทียบกับวิธี IAAC1 สำหรับไฟล์เอกสารภาษาอังกฤษ

การใช้เทคนิคเหล่านี้จะมีความไม่แตกต่างกับ IAAC1 เลย เมื่อใช้เข้ารหัสกับไฟล์ที่มีการกระจายของค่าความน่าจะเป็นต่างออกไปจาก ค่าความน่าจะเป็นที่ได้จากการเตรียมค่าความถี่เริ่มต้นด้วยการใช้การจัดกลุ่มแบบการแบ่งแยกโดยใช้ค่าเฉลี่ยเค ดังจะเห็นได้จากในการเข้ารหัสกับไฟล์ในคลังข้อมูลแคนเทอเบอร์รีขนาดใหญ่ (ตารางที่ 4.11) การกำหนดให้ค่าความน่าจะเป็นของสัญลักษณ์ ‘space’ และ ‘e’ นั้น ตลอดจนสัญลักษณ์อื่นๆ ให้สอดคล้องกับการกระจายของ

ค่าความน่าจะเป็นในแต่ละไฟล์ใดๆ นั้น นับเป็นสิ่งที่ทำได้ยากที่จะให้เกิดผลที่ดี ต่อค่าอัตราส่วนการบีบอัด ในทุกๆ ไฟล์ที่นำมาเข้ารหัส ดังนั้นในกรณีทั่วไปแล้ว การเข้ารหัสเอกสารภาษาอังกฤษจะสามารถใช้ค่าความถี่เริ่มต้นด้วยการใช้การจัดกลุ่มแบบการแบ่งแยกโดยใช้ค่าเฉลี่ยเค ที่ระดับจำนวนกลุ่มเท่ากับ 14 ($k = 14$) แล้วทำให้ได้ค่าอัตราส่วนการบีบอัดที่ดีขึ้นได้ เมื่อเทียบกับวิธีของ IAAC1

รายการอ้างอิง

[1] Apparaju, R., and Agarwal, S. An Arithmetic Coding Scheme by Converting the Multisymbol Alphabet to M-ary Alphabet. Proceedings of the International Conference on Computational Intelligence and Multimedia Applications (ICCIMA 2007) 04(2007): 142-146.

[2] Nobutaka, K., MANABE, T., and Masahiro, N. Adaptive arithmetic coding for image prediction errors. Circuits and Systems. IEEE International Symposium on Circuits and Systems (ISCAS 2004) 111(2004) : 961-964.

[3] Huang, B., Ahuja, A., Huang, H.-L. et. al. Comparison of Arithmetic Coding and Prefix Coding with the CCSDS Lossless Compression Recommendation for Satellite Data. AMS 13th Conf. on Satellite Meteorology and Oceanography (2004) : 1.14.

[4] Murthy, C., and Mishra, P. Lossless Compression Using Efficient Encoding of Bitmasks. Proceedings of the 2009 IEEE Computer Society Annual Symposium on VLSI (2009) : 163-168.

[5] Powell, M. Evaluating lossless compression methods. (2001).

[6] Rice, R. F. Some practical universal noiseless coding techniques. Part III, Module PSI14,K+ [microform] / Robert F. Rice, Pasadena, Calif. National Aeronautics and Space Administration, Jet Propulsion Laboratory, California Institute of Technology : National Technical Information Service, distributor, 1991.

[7] Eric, B., Malte, C., and Joachim, K. Arithmetic Coding revealed - A guided tour from theory to praxis. McGill University, School of Computer Science, Sable Research Group, (2007).

[8] Burrows, M., and Wheeler, D. J. A block-sorting lossless data compression algorithm. SRC Research Report, Digital Systems Research Center, 1994.

[9] Cleary, J., and Witten, I. Data compression using adaptive coding and partial string matching. IEEE Trans. Commnu 32, 4 (1984) : 396-402.

[10] Huffman, A., A Method for the Construction of Minimum-Redundancy Codes. Proceedings of the Institute of Radio Engineers 40, 9 (1952) : 1098-1101.

[11] Golomb, S. W. Run-length coding. IEEE Transactions on Information Theory 12, 3 (1966) : 399-401.

[12] Witten, I. H., Neal, R. M., and Cleary, J. G. Arithmetic coding for data compression. Communications of the ACM 30, 6 (1987) : 520-540.

[13] Ziv, J., and Lempel, A. A Universal Algorithm for Sequential Data Compression. IEEE Transactions on Information Theory 23, 3 (1977) : 337-343.

[14] Howard, P. G., and Vitter, J. S. Practical Implementations of Arithmetic Coding. : Brown University, 1991.

[15] Robert, L., and Nadarajan R. Simple lossless preprocessing algorithms for text compression. IET Software 3, 1 (2009) : 37- 45.

[16] Soyjaudah, K. M. S., and Ramsamy, S. A comparative study of context free models of arithmetic coding. EUROCON'2001, Trends in Communications, International Conference 2, 2 (2001) : 428-431.

[17] Otten, F., Irwin, B., and Thinyane, H. Evaluating text preprocessing to improve compression on maillogs. Proceedings of the 2009 Annual Research Conference of the South African Institute of Computer Scientists and Information Technologists (2009) : 44-53.

[18] Blelloch, G. E. Introduction to Data Compression. Computer Science Department, Carnegie Mellon University, 2001.

[19] Said, A. Introduction to Arithmetic Coding Theory and Practice. Hewlett-Packard Laboratories Report, 2004.

[20] Kochanek, J., Lansky, J., and Uzel, P. et al. The new statistical compression method: Multistream compression. Applications of Digital Information and Web Technologies (ICADIWT 2008) (2008) : 320 – 325.

[21] Hyoung Joong, K. A new lossless data compression method. Multimedia and Expo, 2009. ICME 2009. IEEE International Conference (June 2009) : 1740-1743.

[22] Hyoung Joong, K. A Fast Implementation of Arithmetic Coding. Web Conference (APWEB), 2010 12th International Asia-Pacific (April 2010) : 419-423.

[23] Barbir, A. A New Fast Approximate Arithmetic Coder. Proceedings of the 28th Southeastern Symposium on System Theory (SSST '96) (1996).

[24] Takamura, S., and Takagi, M. Lossless image compression with lossy image using adaptive prediction and arithmetic coding. Data Compression Conference, 1994. DCC '94. Proceedings (March 1994) : 166-174.

[25] Pfefferman, J. D. On the estimation of the probability distribution of a non stationary source for lossless data compression. Image Processing, International Conference 2 (1997) : 270.

[26] Zhou, F., Yang, R., and Li, B. Probability Estimation in Arithmetic Coding and Its Application. Proceedings of the 11th Joint Conference on Information Sciences (2008) (December 2008) : 1-6.

[27] Isal, R. Y. K., Moffat, A., and Ngai, A. C. H. Enhanced Word-Based Block-Sorting Text Compression. ACSC 4 (2002) : 129-137.

[28] Shu, H., Huang, H., Li, T. et al. Bit-Plane Coding for Source with Generalized Gaussian Distribution. Proceedings of the 2009 11th IEEE International Symposium on Multimedia (2009) : 17-23.

[29] Suwannik, W., and Chongstitvatana, P., Solving Large Scale Problems using Estimation Distribution Algorithm with Arithmetic Coding. Proceedings of International Symposium on Communications and Information Technologies (ISCIT) (October 16-19 2007) : 358-363.

[30] Shannon, E. A mathematical theory of communication. SIGMOBILE Mob. Comput. Commun. Rev. 5, 1 (2001) : 3-55.

[31] Jain, A., Murty, M., and Flynn, P. Data Clustering: A Review. ACM Computing Survey 31, 3 (1999) : 264-323.

[32] MacQueen, J. B. Some Methods for Classification and Analysis of MultiVariate Observations. Proc. of the fifth Berkeley Symposium on Mathematical Statistics and Probability (1967) : 281-297.

[33] Jirong, G., Jieming, Z., and Xianwe, C. An Enhancement of K-means Clustering Algorithm. Business Intelligence and Financial Engineering, 2009. BIFE '09. International Conference (July 2009) : 237-240.

[34] Bradley, P. S., and Fayyad, U. M. Refining Initial Points for K-Means Clustering. Proceedings of the Fifteenth International Conference on Machine Learning (1998) : 91-99.

[35] Weibull, W. A Statistical Distribution Function of Wide Applicability. Journal of Applied Mechanics (1951) : 293 - 299.

[36] Sturges, H. A. The Choice of a Class Interval. Journal of the American Statistical Association 21, 153 (1926) : 65-66.

[37] Doane, D. P. Aesthetic frequency classification. American Statistician 30 (1976) : 181–183.

[38] Wand, M. P. Data-based choice of histogram Bin Width. The American Statistician 51, 1, (February 1997) : 59-64.

[39] Nunez-Yanez, J. L., Chen, X., and Canagarajah, N. et al. Statistical Lossless Compression of Space Imagery and General Data in a Reconfigurable Architecture. Proceedings of the 2008 NASA/ESA Conference on Adaptive Hardware and Systems (2008) : 172-177.

ประวัติผู้เขียนวิทยานิพนธ์

นายอนรรฆพล เวียงพล เกิดเมื่อวันที่ 13 กุมภาพันธ์ 2519 สำเร็จการศึกษาในหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต และปริญญาวิศวกรรมศาสตรมหาบัณฑิต สาขาวิชาวิศวกรรมอุตสาหการ ภาควิชาวิศวกรรมอุตสาหการ จุฬาลงกรณ์มหาวิทยาลัย ในปี พ.ศ. 2540 และ พ.ศ. 2542 ตามลำดับ ต่อมาในปี พ.ศ. 2550 ได้เข้าศึกษาต่อในหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต สาขาวิชาวิศวกรรมคอมพิวเตอร์ ภาควิชาวิศวกรรมคอมพิวเตอร์ จุฬาลงกรณ์มหาวิทยาลัย