รายงานการวิจัย

เครื่องจักรสถานะเชิงน่าจะเป็นสำหรับเรียนรู้ส่วนเพิ่มเพื่อการจำแนกลำดับสัญลักษณ์

จิตรกร พูลโพธิ์ทอง

ทุนวิจัยมหาวิทยาลัยรามคำแหง
ปีงบประมาณ 2559

Research Report

Incremental Learning Probabilistic State Machine for Symbolic Sequence Classification

Jittakorn Pullpothong

Ramkhamhaeng University Research Fund
Fiscal Year 2016

บทคัดย่อ

เรื่อง เครื่องจักรสถานะเชิงน่าจะเป็นสำหรับเรียนรู้ส่วนเพิ่มเพื่อการจำแนกลำดับสัญลักษณ์

ผู้วิจัย จิดรกร พูลโพธิ์ทอง
สาขาวิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์
มหาวิทยาลัยรามคำแหง

ปีงบประมาณ ๒๕๕๙

เทคโนโลยีสารสนเทศที่ก้าวหน้าก่อให้เกิดข้อมูลมากมายมหาศาล เช่น ข้อมูลทางชีวสารสนเทศ ข้อมูลทางภูมิศาสตร์ ข้อมูลทางการแพทย์ ข้อมูลทางการเงิน เป็นต้น จึงจำเป็นจะต้องมีเครื่องจักรสำหรับเรียนรู้ที่สามารถรองรับข้อมูลที่มีปริมาณมากและไม่จำกัดจำนวน เพื่อใช้ในการจำแนกข้อมูลดังกล่าว ข้อมูลเหล่านี้สามารถแทนได้ด้วยลำดับเชิงสัญลักษณ์ซึ่งใช้ในระบบการเรียนรู้ของเครื่องจักรในทางปฏิบัติ ซึ่งการเรียนรู้เพื่อจำแนกลำดับเชิงสัญลักษณ์แต่เดิมนั้นจำกัดความยาวของลำดับเหตุการณ์ ทำให้การตอบสนองต่อการขึ้นอยู่กับช่วงระยะยาว (long range dependence) ของข้อมูลทำได้ไม่ดีพอ และวิธีการเรียนรู้ส่วนใหญ่ตั้งอยู่บนสมมติฐานในการเรียนรู้แบบกลุ่มเพียงครั้งเดียว ไม่สามารถครอบคลุมต่อลักษณะข้อมูลที่มีการเปลี่ยนแปลงอย่างรวดเร็วและตลอดเวลาดังเช่นในปัจจุบัน ดังนั้นงานวิจัยนี้จึงได้นำเสนอแบบจำลองสำหรับการเรียนรู้แบบใหม่ ซึ่งเป็นแบบจำลองสถานะเชิงน่าจะเป็นที่สามารถเรียนรู้ส่วนเพิ่มได้ตลอดเวลา รองรับต่อการขึ้นอยู่กับช่วงระยะยาวโดยใช้หลักการสายอักขระเอกลักษณ์สั้นสุดเป็นตัวตัดคำ และสามารถนำไปประยุกต์ใช้กับระบบที่สามารถตัดสินใจได้ทันกาล โดยมีความซับซ้อนเชิงเวลาและเชิงพื้นที่ในการสร้างแบบจำลองจากสายอักขระความยาว n โดยทำได้ดีที่สุด $O(n)$ และในกรณีที่แย่ที่สุด $O(n^2)$ งานวิจัยนี้ได้ทำการทดลองใช้การเรียนรู้เพื่อจำแนกข้อมูลดีเอ็นเอของแบคทีเรีย อี.โคไล (E. Coli) ที่เป็นกลุ่มที่เป็นโปรโมเตอร์และกลุ่มที่ไม่เป็น โดยมีความแม่นยำสูงถึงร้อยละ 96.23 ผิดพลาดเพียงร้อยละ 3.77 ซึ่งผิดพลาดน้อยที่สุดเมื่อเทียบกับขั้นตอนวิธีอื่นจากงานวิจัยก่อนหน้า

คำสำคัญ: การจำแนกลำดับสัญลักษณ์ ลูกโซ่มาร์คอฟ ออโตมาดาเชิงน่าจะเป็น ภาษาเชิงน่าจะเป็น การเรียนรู้ส่วนเพิ่ม

Abstract

Research Title: Incremental Learning Probabilistic State Machine for Symbolic Sequence Classification

Researcher: Jittakorn Pullpothong

Department of Computer Engineering, Faculty of Engineering, Ramkhamhaeng University

Fiscal Year: 2016

The advance of Information technology today cause enormous data stream such as bioinformatic data, geological data, medical data, and financial data etc. Machine learning system to support continuous and unlimited data is required for classifying those enormous data. These data can be represented by symbolic sequences for machine learning system in practical. However, most of former works use limitation of length of training sequences to limit unbounded memory which cause them are not able to cover long range dependency sufficiently. Also, their batch learning methods are hard to deal with variant and rapid changeable data occurring in these day. In this research, we present a novel learning machine which is the probabilistic finite state machine to support incremental learning and long range dependency by using minimum unique substrings as the key. The time and space complexity of the best case is $O(n)$ and the worse case is $O(n^2)$ while n is a length of training data. The experiment of classifying DNA sequences of promoter and non-promoter E. coli bacteria show the accuracy of classification 96.23% and the error only 3.77% which is the best compared to former works.

Keywords: symbolic sequence classification, Markov chain, probabilisitic finite automata, probabilistic languages, incremental learning.

กิตติกรรมประกาศ

ขอขอบพระคุณผู้มีส่วนร่วมในความสำเร็จทุกท่าน ได้แก่ บิดา มารดา ภรรยา พ่อแม่ครูอาจารย์ ญาติ มิตร ที่คอยเป็นกำลังใจ เพื่อนร่วมงานทั้งคณาจารย์และเจ้าหน้าที่ของมหาวิทยาลัยรามคำแหงที่ช่วยอำนวยความสะดวกและช่วยชี้แนะ นักวิทยาการคอมพิวเตอร์ทุกท่านที่ได้สร้างผลงานวิจัยก่อนหน้าอันเป็นความรู้พื้นฐานให้งานวิจัยนี้ และท้ายที่สุดคือมหาวิทยาลัยรามคำแหงที่ให้ทุนสนับสนุนงานวิจัยในครั้งนี้ ให้สำเร็จลุล่วงไปด้วยดี

จิดรกร พูลโพธิ์ทอง
ผู้วิจัย

สารบัญ

สารบัญรูป

รูปที่ หน้า

สารบัญตาราง

ตารางที่ หน้า

บทที่ 1
บทนำ

1.1 บทนำ

ปัจจุบันเทคโนโลยีสารสนเทศ (information technology) ได้ถูกพัฒนาไปอย่างก้าวหน้า ก่อให้เกิดการผลิตข้อมูลเหตุการณ์ (events) ในชีวิตประจำวันของผู้คนจำนวนมหาศาลในรูปแบบของข้อมูลดิจิตอล ไม่ว่าจะเป็น ภาพ เสียง ตัวอักษร รวมถึงข้อมูลที่ใช้ในทางวิทยาศาสตร์ เช่น ข้อมูลคลื่นไฟฟ้าหัวใจ (ECG: electrocardiography) และข้อมูลชีวสารสนเทศ (bioinformatics information) เช่น ข้อมูลรหัสพันธุกรรม (genetic code) ทำให้เกิดสภาวะข้อมูลจำนวนมหาศาลหรือเรียกว่าบิ๊กเดต้า (big data) อย่างไรก็ตาม ข้อมูลจำนวนมหาศาลเหล่านี้สามารถนำมาวิเคราะห์หาความจริงของปรากฏการณ์ต่างๆที่แฝงอยู่ในข้อมูล เพื่อให้ประโยชน์ต่อมนุษย์ เช่น การหาความสัมพันธ์ของลำดับรหัสดีเอ็นเอสำหรับจำแนกลักษณะเฉพาะของสิ่งมีชีวิต การหารูปแบบของคลื่นไฟฟ้าหัวใจที่แสดงความผิดปกติของร่างกาย การหารูปแบบการใช้งานข้อมูลทางเครือข่ายอินเทอร์เน็ตที่แสดงถึงการโจมตีหรือบุกรุกเครือข่าย เป็นต้น การวิเคราะห์ข้อมูลที่มีปริมาณมาก จำเป็นจะต้องมีขั้นตอนวิธีสำหรับเรียนรู้ข้อมูลเพื่อใช้ทำนายและจำแนกประเภทข้อมูล โดยอาศัยโครงสร้างที่เหมาะสมเพื่อจัดเก็บข้อมูลจากการเรียนรู้ และนับจำนวนในทางสถิติที่จำเป็นต่อการตัดสินใจสำหรับปัญหาต่างๆ

ข้อมูลเหตุการณ์ต่างๆตามที่ได้กล่าวข้างต้นนั้นแบ่งออกเป็น 2 ประเภทคือ ข้อมูลเชิงสัญลักษณ์ (symbolic data) สำหรับแทนข้อมูลที่ไม่มีการคำนวณเชิงตัวเลข เช่น แทนสีแดง เขียว และน้ำเงินด้วยสัญลักษณ์ R, G, B ตามลำดับ และข้อมูลเชิงตัวเลข (numeric data) สำหรับแทนข้อมูลที่ถูกนำไปใช้ในการคำนวณเชิงตัวเลข เช่น ความเร็ว 50.5 กิโลเมตรต่อชั่วโมง แทนด้วย v = 50.5 เป็นต้น อย่างไรก็ดี ข้อมูลทั้ง 2 แบบนั้นอาจอยู่ในรูปแบบลำดับของเหตุการณ์ที่เกิดขึ้น (sequence of events) แทนได้ด้วย $x_1, x_2, x_3, \ldots, x_k$ หรือเทียบได้กับสายอักขระ $x_1x_2x_3\ldots x_k$ เมื่อ x_i แทนเหตุการณ์ลำดับที่ i และ n คือจำนวนเหตุการณ์ที่สังเกตุได้ตามคาบเวลา ซึ่งในงานวิจัยนี้จะเน้นไปที่ขั้นตอนวิธีที่ใช้กับข้อมูลลำดับเหตุการณ์เชิงสัญลักษณ์ ตัวอย่างเช่น ข้อมูลรหัสดีเอ็นเอประกอบไปด้วยสัญลักษณ์ เรียกว่า นิวคลีโอไทด์ (nucleotide) แทนได้ด้วยสัญลักษณ์ A, C, G, T ประกอบกันขึ้นเป็นลำดับ แทนได้ด้วยสายอักขระตัวอย่าง ACCGTAAGCTTT... เป็นต้น แต่สำหรับข้อมูลลำดับเหตุการณ์เชิงตัวเลขหรือข้อมูลอนุกรมเวลา (time series) นั้นก็สามารถนำมาเปลี่ยนเป็นข้อมูลลำดับเหตุการณ์เชิงสัญลักษณ์ด้วยการแบ่งนับ (quantization) เพื่อใช้ขั้นตอนวิธีเดียวกันกับข้อมูลเชิงสัญลักษณ์ได้เช่นกัน (Lin, Keogh, Lonardi, &Chiu, 2003)

ดังที่ได้กล่าวไปว่าข้อมูลในปัจจุบันนั้นมีปริมาณมากมายมหาศาล โดยเฉพาะข้อมูลรหัสดีเอ็นเอ โดยรหัสดีเอ็นเอของมนุษย์มีความยาวถึงพันล้านตัวอักษร รหัสดีเอ็นเอของแบคทีเรียบางชนิดมีความยาวถึงสิบล้านตัวอักษร การนำข้อมูลเหล่านี้มาใช้ประโยชน์สำหรับวิเคราะห์หากฎเกณฑ์ที่แฝงอยู่ในข้อมูล เช่น ค้นหาสายอักขระของยีน[1] (gene) การนับจำนวนสายอักขระย่อย การหาแบบรูป (pattern) ของสายอักขระยีน เพื่อใช้ในการเปรียบเทียบ จำแนกข้อมูล จำเป็นต้องใช้ขั้นตอนวิธีการเรียนรู้ที่มีโครงสร้างข้อมูลและขั้นตอนวิธีเพื่อจัดเก็บและจัดการข้อมูลอย่างมีประสิทธิภาพทั้งเวลาและหน่วยความจำ ยกตัวอย่างเช่น การค้นหาสายอักขระยีนที่เกิดซ้ำในรหัสดีเอ็นเอ การเปรียบเทียบยีน (gene) การจำแนกกลุ่มยีนที่ควบคุมพันธุกรรมที่คล้ายกัน เป็นต้น อย่างไรก็ดีเนื่องจากลักษณะของข้อมูลในปัจจุบันและในอนาคตนั้นอาจจะมีข้อมูลส่วนเพิ่ม (incremental data) ที่ถูกผลิตขึ้นมาใหม่อย่างต่อเนื่อง เช่น ข้อมูลดีเอ็นเอ ข้อมูลคลื่นไฟฟ้าหัวใจ ข้อมูลภาษาธรรมชาติ (natural language) ข้อมูลทางการเงิน เป็นต้น การเรียนรู้ของเครื่องจักรแบบเดิมซึ่งรวบรวมข้อมูลแล้วส่งให้เครื่องจักรทำการเรียนรู้เพื่อใช้งานระยะยาวในครั้งเดียว หรือเรียกว่า การเรียนรู้แบบกลุ่ม (batch learning) ซึ่งตั้งอยู่บนข้อสันนิษฐานที่ว่าข้อมูลที่เรียนรู้นั้นจะไม่เปลี่ยนแปลงอีก อาจจะไม่เพียงพอในการรองรับข้อมูลที่เพิ่มขึ้นและมีปริมาณมากอีกทั้งยังมีการเปลี่ยนแปลงที่รวดเร็วตามทิศทางการใช้งานในปัจจุบัน และในบางปัญหานั้นอาจจะมีความจำเป็นในการเฝ้าระวังและต้องการการตัดสินใจอย่างรวดเร็วพร้อมกันไปกับการเรียนรู้ เช่น การเฝ้าระวังอาการของคนไข้ หรือ การเฝ้าระวังสัญญาณแผ่นดินไหว เป็นต้น ทำให้งานวิจัยอันเกี่ยวกับการเรียนรู้และจัดการกับข้อมูลนั้นจะต้องหันมาพิจารณา ขั้นตอนวิธีและโครงสร้างข้อมูลที่สามารถรองรับข้อมูลส่วนเพิ่มที่เกิดขึ้นอย่างไม่จำกัดด้วยอีกทางหนึ่ง หรือเรียกว่า การเรียนรู้ส่วนเพิ่ม (incremental learning) (Ade & Deshmukh, 2013) หรือเรียกอีกอย่างหนึ่งว่า การเรียนรู้ของเครื่องจักรแบบเชื่อมตรง (online machine learning) ซึ่งเป็นขั้นตอนวิธีการเรียนรู้จากข้อมูลที่เกิดขึ้นไปตามลำดับไม่จำกัดจำนวน โดยไม่ต้องรวบรวมข้อมูลสำหรับเรียนรู้ในครั้งเดียว รวมถึงต้องพิจารณาเรื่องความรวดเร็วในการตัดสินใจของเครื่องจักรเรียนรู้ โดยปัจจุบันงานวิจัยที่เกี่ยวข้องกับการเรียนรู้ส่วนเพิ่มนั้นส่วนใหญ่จะมุ่งไปในทางของข้อมูลลำดับเชิงตัวเลข ด้วยหลักการหาค่าที่ดีที่สุดทางสโทแคสติก (Geppert & Hammer, 2016) เช่น เครื่องจักรเวกเตอร์สนับสนุน (support vector machine) และ การแพร่กระจายย้อนกลับ (back propagation) ถึงกระนั้นการเรียนรู้ส่วนเพิ่มในปัจจุบันก็ยังมีกระบวนทัศน์ (learning paradigm) ในรายละเอียดที่ยังไม่ชัดเจนสักเท่าไร และยังมีความแม่นยำในการใช้งานไม่สูงมากนักเนื่องจากมีข้อจำกัดทางขั้นตอนวิธีและหน่วยความจำที่ใช้ อีกทั้งยังไม่สามารถนำผลของการเรียนรู้ไปแปลผลสำหรับใช้วิเคราะห์ผลอย่างอื่นได้ เนื่องจากแบบจำลองดังกล่าวใช้สถิติล้วนโดยไม่ได้เก็บกฎเกณฑ์ความสัมพันธ์ระหว่างคำหรือไวยากรณ์เอาไว้

---

[1] ยีน หมายถึง ส่วนหนึ่งของสายดีเอ็นเอ (DNA segment)

เพื่อให้สอดคล้องกับลักษณะข้อมูลลำดับเชิงสัญลักษณ์ที่มีจำนวนมากในปัจจุบันและเหตุผลตามข้างต้น งานวิจัยนี้จึงสนใจขั้นตอนวิธีการเรียนรู้ส่วนเพิ่มโดยใช้ข้อมูลลำดับเชิงสัญลักษณ์ในการเรียนรู้โดยใช้หลักการภาษาเชิงน่าจะเป็น ซึ่งเป็นทางเลือกที่น่าสนใจที่จะสามารถนำผลของการเรียนรู้มาแปลผลในรูปของสายอักขระได้ ซึ่งมีรูปแบบของไวยากรณ์ของข้อมูลผสมผสานกับสถิติ โดยหลักการเรียนรู้จะคำนึงว่าโครงสร้างที่เก็บสายอักขระเรียนรู้ควรเป็นอย่างไรและมีการปรับพารามิเตอร์อย่างไรจึงจะทำให้การเรียนรู้เป็นไปได้อย่างมีประสิทธิภาพ กล่าวคือทั้งหน่วยความจำที่ใช้และเวลาที่ใช้ในการเรียนรู้จะต้องไม่เกินฟังก์ชันพหุนาม และจะต้องมีความถูกต้องแม่นยำในการใช้งานที่ดีอย่างมีหลักการรองรับ วิธีการที่ใช้สำหรับเรียนรู้ข้อมูลสายอักขระโดยใช้การนับความน่าจะเป็นที่ได้รับความนิยมได้แก่ แบบจำลองภาษาเชิงน่าจะเป็น (probabilistic language model) ตัวอย่างเช่น ออโตมาตาเชิงน่าจะเป็น (probabilistic finite automata)(Vidal, Thollard, Higuera &Casacuberta, 2005) แบบจำลองเอ็นแกรม (n-gram) (Cavnar & Trenkle, 1994) แบบจำลองมาร์คอฟแบบซ่อน (hidden Markov model) (Rabiner, 1989) แบบจำลองออโตมาตาคำเติมท้ายเชิงน่าจะเป็น (probabilisitc suffix automata) (Ron, Singer, &Tishby, 1996) เป็นต้น ถึงกระนั้น แบบจำลองเหล่านี้ตั้งอยู่บนสมมติฐานการเรียนรู้ของสายอักขระที่มีความยาวจำกัด โดยใช้กระบวนการเรียนรู้จากสายอักขระที่จำกัดความยาว n เพื่อลดจำนวนความยาวในกรณีที่สายอักขระมีความยาวไม่จำกัด ให้อยู่ในมิติของข้อมูลที่สามารถจัดการได้ แล้วนำหลักการของลูกโซ่มาร์คอฟ (Markov chain) ความยาว n สำหรับคำนวณหาความน่าจะเป็น แต่หากพิจารณาในกรณีที่ข้อมูลอาจมีเพิ่มขึ้นได้ตลอดเวลาแล้ว ปัญหาคือความยาว n เท่าไรจึงจะเหมาะสมต่อสภาพข้อมูล และปัญหาอีกประการหนึ่งคือหากใช้ในการเรียนรู้ข้อมูลส่วนเพิ่มนั้นอาจมีผลกระทบต่อโครงสร้างเดิม ทำให้ไม่สามารถเรียนรู้เพิ่มเติมได้อย่างมีประสิทธิภาพ แต่ข้อดีของการใช้แบบจำลองภาษาเชิงน่าจะเป็นนี้คือ สามารถสร้างด้วยแบบจำลองเครื่องจักรสถานะเชิงน่าจะเป็น (probabilistic state machine) เช่น ออโตมาตาเชิงน่าจะเป็นที่สามารถดำเนินการคำนวณได้ในแบบเวลาจริง (real time) โดยสามารถอ่านอินพุตแต่ละตัวแล้วเปลี่ยนสถานะไปเพื่อคำนวณผลลัพธ์ได้ทันที และออโตมาตาก็เป็นแบบจำลองที่รู้จักกันโดยทั่วไปอย่างกว้างขวางในสาขาวิทยาการคอมพิวเตอร์จึงทำความเข้าใจได้ไม่ยาก แม้ว่ากลุ่มแบบจำลองภาษาเชิงน่าจะเป็นจะใช้งานได้ดีระดับหนึ่งในทางปฏิบัติ แต่ประเด็นสำคัญคือ หากเหตุการณ์ต่างๆที่เกิดขึ้นมีความสัมพันธ์ระหว่างข้อมูลยาวกว่าขนาด n ที่จำกัดไว้ หรือที่เรียกว่าข้อมูลมี การขึ้นอยู่กับช่วงระยะยาว (long range dependence) ก็จะทำให้การคำนวณหาความน่าจะเป็นมีความคลาดเคลื่อนได้มากหากมีการเรียนรู้ส่วนเพิ่มในระยะยาว การนำสมมติฐานที่จำกัดความยาวไปใช้งานด้วยการทำนายหรือจำแนกข้อมูลขนาดใหญ่อาจจะทำให้ร้อยละของความถูกต้องลดลง เนื่องจากพบว่าข้อมูลขนาดใหญ่มีความยาวไม่จำกัดนั้นมีการขึ้นอยู่กับช่วงระยะยาว เช่น ภาษาของมนุษย์

และข้อมูลรหัสดีเอ็นเอ สมมติฐานในของแบบจำลองภาษาเชิงน่าจะเป็นดังงานวิจัยที่ผ่านมานั้น อาจจะไม่เพียงพอที่จะรองรับปัญหาการขึ้นอยู่กับช่วงระยะยาวได้ อย่างไรก็ตาม ดังที่กล่าวแล้วว่าปัญหาก็คือการตัดช่วงระยะยาวของคำเท่าไรจึงจะเหมาะสมต่อการเรียนรู้ที่ดีและไม่ทำให้ความสัมพันธ์ของข้อมูลคลาดเคลื่อนมากเกินไป จึงมีความจำเป็นที่จะต้องมีแบบจำลองที่รองรับการขึ้นอยู่กับช่วงระยะยาวของข้อมูลในระดับหนึ่งที่รับประกันได้ว่าไม่คลาดเคลื่อนมากเกินไป นอกจากนั้นแล้วแบบจำลองภาษาเชิงน่าจะเป็นในปัจจุบัน ส่วนใหญ่ยังมีข้อจำกัดในการรับข้อมูลส่วนเพิ่มเข้ามาเรียนรู้ใหม่ซึ่งกระทบกับโครงสร้างเดิม การเรียนรู้ข้อมูลใหม่จึงยังต้องใช้การรวบรวมข้อมูลเก่าแล้วทำการเรียนรู้ใหม่ซึ่งไม่ตอบสนองต่อข้อมูลจำนวนมหาศาลในปัจจุบัน

จากแบบจำลองที่กล่าวมาข้างต้นสรุปได้ว่า แบบจำลองภาษาเชิงน่าจะเป็น เช่น ออโตมาตาเชิงน่าจะเป็นนั้นเหมาะสำหรับการนับความน่าจะเป็นของการเกิดสายอักขระได้ในเวลาจริง สามารถหาความน่าจะเป็นของการเกิดอักขระแต่ละตัว ซึ่งมีประโยชน์ต่อระบบที่ต้องการดำเนินการภายในระยะเวลาที่รวดเร็วทันต่อเหตุการณ์ เช่น ระบบตรวจสอบการเต้นของหัวใจของผู้ป่วย ระบบการควบคุมการขับรถ เป็นต้น แม้ว่าแบบจำลองออโตมาตาเชิงน่าจะเป็นจะมีข้อดีดังกล่าว แต่ในการนำไปใช้งานนั้นอาจจะเกิดความคลาดเคลื่อนในการนับความน่าจะเป็นสำหรับสายอักขระที่มีการขึ้นอยู่กับช่วงระยะยาว และมีข้อจำกัดในการเรียนรู้ข้อมูลส่วนเพิ่ม งานวิจัยนี้จึงได้มีความสนใจว่า ทำอย่างไรจึงจะสามารถสร้างแบบจำลองการเรียนรู้ในรูปของภาษาเชิงน่าจะเป็นเพื่อที่จะสามารถใช้สำหรับใช้เรียนรู้และจัดการกับข้อมูลขนาดใหญ่ โดยสามารถเรียนรู้ส่วนเพิ่มได้ไม่จำกัด มีการคำนึงถึงการขึ้นอยู่กับช่วงระยะยาวตามบริบทของข้อมูล สามารถนำผลการเรียนรู้นั้นมาใช้งานวิเคราะห์ผลที่หลากหลายได้ โดยมีกระบวนทัศน์ในการเรียนรู้ส่วนเพิ่มที่ไม่จำกัดหน่วยความจำ อันจะทำให้สามารถใช้ในการจำแนกกลุ่มข้อมูลได้อย่างแม่นยำและมีประสิทธิภาพทั้งเวลาและหน่วยความจำในการเรียนรู้ เพื่อให้ได้แบบจำลองการเรียนรู้ที่สามารถรองรับสถานการณ์ข้อมูลที่มีปริมาณมากมายมหาศาลที่เกิดขึ้น

ในงานวิจัยนี้จึงได้ทำการศึกษา ค้นคว้า ให้ได้มาซึ่งขั้นตอนวิธีและโครงสร้างข้อมูลแบบใหม่ให้มีประสิทธิภาพคือ สามารถเก็บตัวอย่างสายอักขระที่แตกต่างและสามารถนับจำนวนการเกิดของสายอักขระเหล่านั้น รวมถึงความสามารถในการคำนวณความน่าจะเป็นของสายอักขระที่แตกต่างได้ในแบบเวลาจริง และรองรับการเรียนรู้ข้อมูลส่วนเพิ่มที่เข้ามาในภายหลังได้โดยไม่ต้องล้างโครงสร้างเดิมและไม่จำเป็นต้องรวบรวมข้อมูลเก่าเพื่อกลับมาเรียนรู้ใหม่ โดยใช้โครงสร้างข้อมูลและการคำนวณความน่าจะเป็นด้วยแบบจำลองสถานะเชิงน่าจะเป็นและหลักการตัดคำด้วยสายอักขระเอกลักษณ์ของข้อมูล ซึ่งเป็นสายอักขระที่เกิดขึ้นครั้งเดียวในข้อมูลและสายอักขระเกิดซ้ำ เสนอทฤษฎีโดย Ilie (Ilie & Smith, 2011) เพื่อให้จำนวนสถานะของแบบจำลองมีการเติบโตขณะเรียนรู้ไม่มากเกินไปจนเกินจัดการได้ การวิจัยได้แบ่งออกเป็นสองส่วน คือ การค้นคว้าหาขั้นตอนวิธีและโครงสร้างเพื่อใช้สำหรับการเรียนรู้จากข้อมูลขนาดใหญ่ตาม

วัตถุประสงค์ที่ต้องการ และอีกส่วนหนึ่งคือ การค้นคว้าหาวิธีการในการนำโครงสร้างข้อมูลนั้นไปใช้งานเพื่อจำแนกสายอักขระแล้วทำการทดลองถึงประสิทธิภาพในการใช้งานนั้น ในงานวิจัยส่วนแรกนั้นจะใช้หลักการและทฤษฎีสำหรับพิสูจน์ว่าขั้นตอนวิธีและโครงสร้างข้อมูลที่ได้นั้นเป็นไปตามหลักการ โดยพบว่าขั้นตอนวิธีในการเรียนรู้แบบจำลองในงานวิจัยนี้ใช้ประสิทธิภาพเชิงเวลาและประสิทธิภาพเชิงพื้นที่ภายในดีที่สุด $O(n)$ และแย่ที่สุด $O(n^2)$ เมื่อ n คือความยาวของข้อมูลที่ใช้ในการเรียนรู้ทั้งหมด และในงานวิจัยส่วนที่สองคือ การนำไปใช้งานสำหรับจำแนกข้อมูลลำดับเชิงสัญลักษณ์นั้น จะใช้หลักการคำนวณความน่าจะเป็นของสายอักขระด้วยลูกโซ่มาร์คอฟ และการปรับเรียบ (smoothing) โดยทดลองนำไปใช้งานในการจำแนกกลุ่มข้อมูลดีเอ็นเอของแบคทีเรียชนิดอี.โคไล (E. Coli) โดยแบ่งออกเป็น 2 กลุ่ม กลุ่มหนึ่งเป็นแบบโปรโมเตอร์ (promoter) จำนวน 53 สายและอีกกลุ่มหนึ่งแบบไม่เป็นโปรโมเตอร์ (non-promoter) จำนวน 53 สาย แต่ละสายมีความยาว 57 ตัวอักษร กลุ่มของโปรโมเตอร์จะมีแบบรูปข้อมูลสายอักขระย่อยที่ขึ้นอยู่กับช่วงระยะยาวซึ่งเหมาะสมต่อการทดสอบประสิทธิภาพงานวิจัยนี้ การทดลองจำแนกกลุ่มโดยคำนวณความน่าจะเป็นที่หาได้จากแบบจำลองและวิธีการในงานวิจัยนี้ โดยทำการเปรียบเทียบกับวิธีการที่มีประสิทธิภาพอันเป็นที่ยอมรับและเป็นที่นิยม เช่น ระบบเรียนรู้ของเครื่องจักรแบบผสม(hybrid machine learning) (Towell, Shavik, &Noordewier, 1990) โครงข่ายประสาทเทียมมาตรฐาน (neural network) และ เอ็นแกรม (n-gram) เป็นต้น พบว่าวิธีการในงานวิจัยนี้สามารถจำแนกกลุ่มของแบคทีเรียได้ถูกต้องได้ถึง 102 สายใน 106 สาย ผิดพลาดเพียง 4 สาย ซึ่งเทียบเป็นร้อยละได้ 96.22% เทียบความแม่นยำได้สูงสุดเทียบเท่ากับระบบเรียนรู้ของเครื่องจักรแบบผสม ในขณะที่โครงข่ายประสาทเทียมแบบมาตรฐานได้ความถูกต้องเพียง 98 สายคิดเป็น 92.45% และเอ็นแกรมทำได้ดีที่สุดโดยเลือกใช้ 3-แกรม ถูกต้องเพียง 93 สายคิดเป็น 87.74% จึงสรุปได้ว่า แบบจำลองการเรียนรู้ด้วยโครงสร้างข้อมูลและขั้นตอนวิธีจากงานวิจัยนี้สามารถใช้จำแนกกลุ่มของข้อมูลประเภทลำดับเชิงสัญลักษณ์ได้ผลเป็นที่น่าพอใจอย่างมากทั้งความเร็วในการเรียนรู้และการใช้งานและมีความแม่นยำสูงสุดเมื่อเทียบกับแบบจำลองอันเป็นที่ยอมรับอื่นๆ ซึ่งจะสามารถนำไปประยุกต์ใช้ให้เกิดประโยชน์ในงานวิจัยอื่นๆและโปรแกรมประยุกต์ทางปัญญาประดิษฐ์ได้ต่อไป เนื้อหาในรายงานวิจัยนี้จะกล่าวถึงทฤษฎีที่เกี่ยวข้องซึ่งเป็นพื้นฐานที่จำเป็นในการศึกษาแบบจำลองในงานวิจัยนี้ในบทที่ 2 และแบบจำลองและขั้นตอนวิธีการเรียนรู้และทฤษฎีที่นำเสนอในงานวิจัยนี้จะกล่าวรายละเอียดในบทที่ 3 โดยแสดงผลการทดลองยืนยันทฤษฎีและวิเคราะห์ผลในบทที่ 4 และสรุปผลการวิจัยและข้อเสนอแนะในบทที่ 5

1.2 วัตถุประสงค์

1.2.1 เพื่อให้ได้มาซึ่งขั้นตอนวิธีการเรียนรู้ของเครื่องจักรที่สามารถประยุกต์ใช้กับ ข้อมูลประเภทลำดับสัญลักษณ์ขนาดใหญ่ ทั้งความยาวจำกัดหรือไม่จำกัดได้

1.2.2 เพื่อให้ได้มาซึ่งขั้นตอนวิธีการเรียนรู้ของเครื่องจักรที่สามารถรองรับการเรียนรู้ส่วนเพิ่มของข้อมูลประเภทลำดับสัญลักษณ์

1.2.3 เพื่อให้ได้มาซึ่งแบบจำลองที่สามารถทำนาย หรือจำแนกข้อมูลลำดับสัญลักษณ์ขนาดใหญ่ ทั้งความยาวจำกัดหรือไม่จำกัดได้

1.2.4 เพื่อศึกษาค้นคว้าและทดลองให้ทราบถึง ปัญหา อุปสรรค แนวทางแก้ไขและแนวทางปรับปรุงการเรียนรู้ของเครื่องจักรให้มีความแม่นยำและสะดวกมากขึ้น

1.2.5 เพื่อรองรับงานวิจัยต่อยอดสำหรับนำไปประยุกต์ใช้

1.3 ขอบเขตการวิจัย

1.3.1 ข้อมูลที่ใช้สำหรับวิจัยมุ่งเน้นวิธีการเฉพาะสำหรับข้อมูลลำดับสัญลักษณ์เท่านั้น

1.3.2 ข้อมูลที่ใช้สำหรับวิจัยสามารถใช้ได้ทั้งลำดับสัญลักษณ์ความยาวจำกัดและไม่จำกัด ทั้งรูปแบบความยาวต่อเนื่องและไม่ต่อเนื่อง

1.3.3 ผลงานที่ได้จากการวิจัยได้แก่ ขั้นตอนวิธีในการเรียนรู้ แบบจำลองที่ใช้ในการเรียนรู้ ปัญหา ข้อจำกัด ข้อได้เปรียบและข้อเสียเปรียบ และทฤษฎีที่ได้จากการวิจัย ไม่รวมไปถึงวิธีการประยุกต์ใช้งานจริงในปัญหาต่างๆ

1.3.4 แสดงการพิสูจน์ความถูกต้องของทฤษฎี โดยใช้ทั้งวิธีการพิสูจน์โดยทฤษฎีและการทดลองจากตัวอย่างบางประเภท

1.3.5 แสดงผลความแม่นยำในการจำแนกข้อมูลของแบบจำลองและวิธีการที่ได้จากงานวิจัย โดยใช้วิธีวิเคราะห์เปรียบเทียบกับขั้นตอนวิธีและแบบจำลองที่ได้รับความเชื่อถือเป็นสากล เช่น แบบจำลองเอ็นแกรม แบบจำลองมาร์คอฟแบบซ่อน แบบจำลองออโดมาตาเชิงน่าจะเป็น หรือแบบจำลองอื่นๆ ที่เกี่ยวข้อง อย่างใดอย่างหนึ่งหรือหลายอย่าง เพื่อแสดงประสิทธิภาพอย่างพอเพียง

1.3.6 แสดงประสิทธิภาพในการเรียนรู้และจำแนกข้อมูล แบ่งออกเป็นเชิงเวลาในการเรียนรู้และหน่วยความจำในการเรียนรู้ กำหนดให้ไม่เกินกรอบเวลาเป็นฟังก์ชันพหุนามเมื่อเทียบกับปริมาณข้อมูลที่ใช้ในการเรียนรู้

1.4 ประโยชน์ที่คาดว่าจะได้รับ

1.4.1 สามารถนำผลงานวิจัยไปใช้ในการจำแนกหรือทำนายข้อมูลลำดับสัญลักษณ์ เช่น จำแนกข้อมูลเอกสาร (document classification) จำแนกข้อมูลดีเอ็นเอ (DNA classification) การตรวจสอบสุขภาพของผู้ป่วย (patient's health detection) การตรวจสอบสิ่งผิดปรกติ (anomaly detection), การวิเคราะห์ชีวสารสนเทศ (bioinformation analysis) การทำนายลำดับเหตุการณ์ต่างๆ (events prediction)

1.4.2 สามารถนำผลงานวิจัยไปใช้ในการหาเอกลักษณ์ของข้อมูลที่เกิดขึ้นซ้ำ เช่น ข้อมูลที่เกิดขึ้นซ้ำที่ยาวหรือสั้นที่สุด ข้อมูลที่เกิดขึ้นซ้ำที่มีความถี่ในการเกิดขึ้นมากหรือน้อยที่สุด ซึ่งสามารถนำไปประยุกต์ใช้กับการหาเอกลักษณ์ของข้อมูล

1.4.3 สามารถทราบถึงข้อได้เปรียบและข้อเสียเปรียบของการประยุกต์ใช้ขั้นตอนวิธีและแบบจำลองที่ได้จากงานวิจัย เมื่อเปรียบเทียบกับงานวิจัยอื่น

1.4.4 สามารถทราบถึง ปัญหา ข้อจำกัดในการใช้งาน แนวทางแก้ปัญหา และแนวทางปรับปรุงประสิทธิภาพในการประยุกต์ใช้ทฤษฎีที่ได้จากผลงานวิจัย หรือประยุกต์ใช้กับการเรียนรู้ในข้อมูลที่มีลักษณะที่คล้ายกัน

บทที่ 2
ทฤษฎีที่เกี่ยวข้อง

2.1 บทนำ

งานวิจัยนี้ได้ทำการสำรวจบทความวิจัยซึ่งมีงานวิจัยที่น่าสนใจที่ได้กล่าวถึงทฤษฎีพื้นฐานและการทดลองที่นำมาร่วมใช้ทดสอบและอ้างอิงในงานวิจัยนี้ รวมถึงเป็นงานที่จุดประกายแง่คิดให้งานวิจัยนี้ โดยคัดเลือกมาบางงานวิจัยที่น่าสนใจ ซึ่งบางเรื่องนั้นจะเป็นพื้นฐานให้กับการศึกษาทฤษฎีของงานวิจัยนี้ในบทถัดไป โดยเริ่มจากบทความที่สำรวจงานวิจัยเกี่ยวกับการจำแนกข้อมูลลำดับว่ามีกี่ประเภทและแต่ละประเภทมีข้อดีข้อเสียอย่างไร ประเด็นที่ต้องพิจารณาเกี่ยวกับการจำแนกข้อมูลลำดับมีอะไรบ้าง งานวิจัยการจำแนกข้อมูลช่วงต้น กล่าวถึงความสำคัญ วิธีการทดลองและแก้ปัญหา ซึ่งแสดงถึงปัญหาในปัจจุบันที่น่าสนใจ และงานวิจัยเกี่ยวกับสายอักขระเอกลักษณ์และสายอักขระเกิดซ้ำซึ่งเป็นหลักการที่สัมพันธ์กันกับงานวิจัยนี้ และงานสำรวจบทความวิจัยวิธีการเรียนรู้แบบจำลองสถานะเชิงน่าจะเป็น ซึ่งแสดงถึงทฤษฎีที่เกี่ยวข้องกับแบบจำลองสถานะเชิงน่าจะเป็น และการเรียนรู้เพื่อสร้างแบบจำลองสถานะเชิงน่าจะเป็นโดยสรุป เนื้อหาในบทนี้จะสรุปความเกี่ยวกับบทความดังกล่าวโดยคร่าวเพื่อให้เป็นพื้นฐานในการทำความเข้าใจถึงความเป็นมาของงานวิจัยนี้ได้

2.2 งานวิจัยที่เกี่ยวข้อง

2.2.1 บทความเรื่อง: a brief survey on sequence classification (Xing, Pei, Keogh, 2010)

บทความนี้ได้สรุปสาระสำคัญของงานวิจัยเกี่ยวกับการจำแนกข้อมูลสายอักขระ โดยได้นิยามปัญหาการจำแนกข้อมูลที่น่าสนใจไว้ว่า "ให้ L เป็นเซตของข้อมูลแต่ละกลุ่มแล้ว ตัวจำแนกข้อมูล C คือ ฟังก์ชันที่จับคู่ข้อมูลอินพุต s ไปเป็น l โดยที่ l คือกลุ่มข้อมูลใน L เขียนนิยามได้คือ $C : s \rightarrow l | l \in L$ โดยมีประเด็นสรุปการสำรวจงานวิจัยได้ดังนี้ คือ

2.2.1.1 ประโยชน์การใช้งานจริงของการจำแนกลำดับข้อมูล

1. การเรียนรู้ฟังก์ชันการทำงานของโปรตีนในข้อมูลดีเอ็นเอ (biological sequence classification)

2. การจำแนกข้อมูลทางด้านสุขภาพ (health informatic classification) เช่น ข้อมูลอนุกรมเวลา (time series) ของคลื่นไฟฟ้าหัวใจ

3. การตรวจจับพฤติกรรมผิดปรกติ (anomaly detection) เช่น การใช้ชุดคำสั่งของผู้ใช้งานที่ผิดปรกติ ซึ่งอาจจะเป็นการบุกรุก

4. การจำแนกเอกสาร (document categories classification) ว่าเอกสารต่างๆอยู่ในหมวดหมู่ใด

5. การจำแนกล็อกการใช้งาน (log classification) ของผู้ใช้งานว่าเป็นการทำงานของโปรแกรมหุ่นยนต์หรือไม่

6. การจำแนกข้อมูลธุรกรรมการเงินในธนาคาร (classifying transaction sequence data in a bank) ว่าเป็นการฟอกเงินหรือไม่

2.2.1.2 ชนิดของอินพุตที่ใช้กับการจำแนกลำดับข้อมูล

1. ลำดับเชิงสัญลักษณ์อย่างง่าย (simple symbolic sequence) เป็นข้อมูลลำดับของเหตุการณ์ หรือสายอักขระ เช่น สายอักขระของนิวคลีโอไทด์ (A, C, G, T) ของรหัสดีเอ็นเอ เช่น ACCGTATT

2. ลำดับเชิงสัญลักษณ์ที่ซับซ้อน (complex symbolic sequence) เป็นข้อมูลสายอักขระหลายมิติ เช่น ข้อมูลอาหารที่รับประทาน (milk, egg), (milk, bread), ...

3. อนุกรมเวลาอย่างง่าย (simple time series) คือ ข้อมูลลำดับของจำนวนจริง ซึ่งสามารถนำไปใช้ในการคำนวณเชิงตัวเลข เช่น (t_0, 0.3), (t_1, 0.4), (t_2, 0.5)...(t_n, 0.7)

4. อนุกรมเวลาหลายตัวแปร (multivariate time series) คือ อนุกรมเวลาที่มีหลายตัวแปรในหนึ่งหน่วยเวลา เช่น (t_0, (0.3, 0.5, 0.9)), (t_1, (0.4, 0.3, 0.6)), (t_2, (0.5, 0.8, 0.7)),..., (t_n, (0.7, 0.9, 0.4))

2.2.1.3 ประเด็นปัญหาที่น่าสนใจเกี่ยวกับการจำแนกลำดับข้อมูลเชิงสัญลักษณ์

การจำแนกลำดับของข้อมูลเชิงสัญลักษณ์นั้นควรพัฒนาไปในทิศทางใดนั้น จำเป็นจะต้องพิจารณาปัญหาที่มีอยู่ในปัจจุบันและแนวโน้มในการปรับปรุงวิธีการ โดยมีประเด็นที่น่าสนใจดังนี้ คือ

1. วิธีการจำแนกลำดับข้อมูลที่นิยมในปัจจุบัน เช่น การใช้ต้นไม้ตัดสินใจและโครงข่ายประสาทเทียม นั้นจำเป็นจะต้องใช้เวกเตอร์ลักษณะเฉพาะ เช่น ข้อมูลอากาศมี <หนาว, แห้ง, ฝน>, <หนาว, ชื้น, แดด> เป็นต้น แต่ลำดับข้อมูลเชิงสัญลักษณ์ไม่มีเวกเตอร์ของลักษณะ

เฉพาะ เนื่องจากเป็นสายอักขระความยาวใดๆ ประเด็นปัญหาคือ จะมีวิธีใดเหมาะสมกับการจำแนกลำดับข้อมูลเชิงสัญลักษณ์

2. การแปลงลำดับเชิงสัญลักษณ์ให้จัดอยู่ในรูปเวกเตอร์ลักษณะเฉพาะนั้น จะต้องกำหนดขนาดมิติด้านความยาวของข้อมูล ตามความยาวของสายอักขระ ซึ่งจะทำให้มีความยาวมากเกินไป ผลก็คือทำให้การประมวลผลใช้เวลาและหน่วยความจำสูง

3. นอกเหนือจากความถูกต้องในการจำแนกข้อมูลแล้ว บางครั้งอาจจะต้องการตัวจำแนกข้อมูลที่สามารถนำไปแปลผลข้อมูลอื่นๆได้ ยกตัวอย่างเช่น การนับจำนวนสายอักขระที่เกิดขึ้นบ่อย หรือ ต้องการจัดเก็บฐานข้อมูลของสายอักขระย่อยที่มีความยาวตามที่ต้องการ เพื่อหาความถี่ของการเกิดเหตุการณ์ต่างๆ

2.2.1.4 วิธีการที่ใช้สำหรับการจำแนกข้อมูล

1. ใช้ลักษณะเฉพาะเป็นพื้นฐาน (feature based classification) เช่น เวกเตอร์ลักษณะเฉพาะ (feature vector) เอ็น-แกรม (n-gram)

2. ใช้การวัดระยะทางระหว่างลำดับเป็นพื้นฐาน (sequence distance based classification) เป็นการวัดระยะห่างของลำดับ 2 ลำดับด้วยวิธีการต่างๆ เช่น เวกเตอร์สนับสนุน (support vector machine) เพื่อนบ้านใกล้สุด k อันดับ (k-nearest neighbors) โดยอาจใช้วิธีการอย่างง่ายในการวัดระยะทาง เช่น ระยะทางของยุคลิด (Euclidean distance) โดยกำหนดให้

$$dist(s,s') = \sqrt{\sum_{i=1}^{l} (s[i] - s'[i])^2}$$

ข้อเสียของวิธีนี้คือ นอกจากได้ผลลัพธ์ในการจำแนกข้อมูลแล้ว ไม่สามารถนำองค์ความรู้ที่ได้จากการเรียนรู้ไปแปลผลอย่างอื่น

3. ใช้แบบจำลองเป็นพื้นฐาน (model based classification) เช่น แบบจำลองมาร์คอฟแบบซ่อน (Hidden Markov Model) แบบจำลองเชิงสถิติอื่นๆ

2.2.1.5 ส่วนขยายของปัญหาการจำแนกข้อมูลลำดับ

นอกจากการจำแนกข้อมูลแบบปรกติตามนิยามแล้ว ยังมีปัญหาที่ขยายมากขึ้นดังนี้คือ

1. การจำแนกข้อมูลลำดับช่วงต้น (early classification) เป็นการหาคำตอบในการจำแนกข้อมูลให้เร็วที่สุดเท่าที่ทำได้ในการอ่านอินพุต โดยไม่จำเป็นต้องอ่านอินพุตทั้งหมดแล้ว

จึงตัดสินใจ ซึ่งมีประโยชน์ต่อ โปรแกรมประยุกต์ที่มีระบบการตรวจจับที่ไวต่อเหตุการณ์ เช่น การตรวจจับการบุกรุก (intrusion detection) และ การตรวจคลื่นหัวใจของผู้ป่วยที่ต้องมีการเฝ้าระวัง การตรวจจับสัญญาณแผ่นดินไหวหรือซึนามิ เป็นต้น

2. การจำแนกข้อมูลลำดับแบบกึ่งผู้สอน (semi-supervised sequence classification) เป็นการใช้ข้อมูลแบบมีการระบุกลุ่มและไม่มีการระบุกลุ่มเข้าร่วมกัน ซึ่งข้อมูลบางประเภทไม่มีการระบุกลุ่มที่ชัดเจน แต่สามารถนำมาใช้ร่วมกับการระบุกลุ่มได้ในการเพิ่มประสิทธิภาพการจำแนก

3 การจำแนกข้อมูลลำดับแบบระบุกลุ่มตามลำดับ (sequence classification with a sequence of label) เป็นการอ่านข้อมูลแต่ละตัวแล้วจำแนกกลุ่มของข้อมูลแต่ละตัว ยกตัวอย่าง เช่น หากให้ประโยคในภาษาธรรมชาติเป็นอินพุต คำแต่ละคำในประโยคจะถูกระบุกลุ่มข้อมูลไปตามลำดับ เช่น เป็น noun, noun phrase, verb เป็นต้น ซึ่งอาจจะจำแนกคำแต่ละคำแบบลำพังหรือ จำแนกเนื่องจากมีความสัมพันธ์กับคำอื่นๆ

2.2.2 บทความเรื่อง: Mining Sequences Classifier for Early Prediction (Xing, Pei, Dong, &Yu, 2008)

งานวิจัยนี้ได้นำเสนอวิธีการจำแนกข้อมูลลำดับช่วงต้น (early prediction) ใช้ สำหรับทำนายข้อมูลที่มีความวิกฤติ เช่น การจำแนกข้อมูลทางการแพทย์ การทำนายเกี่ยวกับภัยพิบัติ เช่น การทำนายซึนามิก่อนจะเกิดเหตุการณ์จริงได้ทันกาลย่อมสามารถรักษาชีวิตคนไว้ได้มากมาย หรือการพบการเต้นหัวใจที่เริ่มผิดปรกติของผู้ป่วย ย่อมสามารถป้องกันเหตุอันตรายที่จะเกิดขึ้นได้ เป็นต้น ตัวจำแนกจะอ่านอินพุตเพียงข้อมูลช่วงต้นแล้วตัดสินใจได้โดยไม่จำเป็นจะต้องอ่านข้อมูลจนครบทั้งหมดแล้วจึงตัดสินใจ โดยวัดค่าสมมติซึ่งเป็นค่าที่แสดงให้เห็นว่า การจำแนกข้อมูลลำดับช่วงต้น มีค่าเท่ากันกับการจำแนกทั้งหมด ซึ่งผู้ใช้งานจะเป็นผู้กำหนดพารามิเตอร์ ρ_0 คือ ค่าความถูกต้องน้อยที่สุดที่ยอมรับ

นิยามที่เกี่ยวข้องมีดังนี้ ให้ลำดับ $s = a_1 \ldots a_l$ และลำดับที่เป็นคำนำหน้าของ s แทนได้ด้วย $s' = a'_1 \ldots a'_l$ โดยที่ $1 \le l' \le l$ หรือเขียน $s' = s[1,l']$ให้ตัวจำแนกข้อมูลลำดับ C ดำเนินการไปตามลำดับทีละตัวแล้ว จะมี l_0 โดยที่

$$C(s[1,l_0]) = C(s[1,l_0+1]) = C(s[1,l_0+k]) = C(s)$$

โดยมีค่า $Cost(C,s) = l_0$ ซึ่ง

$$Cost(C,s[1,l_0]) = Cost(C,s[1,l_0+1]) = \cdots = Cost(C,s)$$

ซึ่งจะได้ว่า $Cost(C,s) \leq |s|$ ซึ่งใช้สำหรับคำนวณประสิทธิภาพ

การคำนวณหาค่าความถูกต้อง ให้มีฐานข้อมูลของลำดับทั้งหมดแทนด้วย SDB และฟังก์ชันคำนวณความถูกต้องสมมติให้ชื่อ $Accuracy$ แล้ว

$$Accuracy(C,SDB) = \frac{|\{C(s) = c | (s,c) \in SDB\}|}{|SDB|}$$

กล่าวอย่างง่ายคือ ค่าความถูกต้องคือจำนวนที่พบว่ามี c ในฐานข้อมูลหารด้วยขนาดของฐานข้อมูล ผู้วิจัยได้เสนอวิธีการ 2 วิธีสำหรับคำนวณหาค่าความถูกต้องดังกล่าว คือ 1.ใช้กฎการจำแนกข้อมูล (a sequence classification rule :SCR) 2. ต้นไม้ตัดสินใจ (Generalized sequenctial decision tree: GSDT)

1. วิธีใช้กฎการจำแนกข้อมูล คือ กฎ $R: f_1 \rightarrow f_2 \rightarrow \cdots \rightarrow f_n \Rightarrow c$ เมื่อ $f_1, f_2, \ldots, f_n$ เป็นลักษณะเฉพาะ (features) และ $c \in L$ เป็นกลุ่มข้อมูลที่จำแนกได้ โดยในงานวิจัยจะกล่าวถึงวิธีการเลือกเซตของกฎเพื่อเป็นตัวจำแนกข้อมูล โดยใช้ลักษณะเฉพาะเป็นตัวสร้างกฎแต่ละกฎขึ้นมา โดยกฎและลักษณะเฉพาะที่เลือกมานั้นจะเป็นตัวกำหนดความสามารถในการจำแนกช่วงต้นของข้อมูล

2. วิธีการใช้ต้นไม้ตัดสินใจ ผู้วิจัยเลือกใช้ลักษณะเฉพาะของข้อมูลเพื่อใช้แทนข้อมูลลักษณะประจำ (attribute) ของต้นไม้ โดยแต่ละบัพของต้นไม้ก็คือลักษณะเฉพาะของข้อมูล โดยเลือกลักษณะเฉพาะที่มีคุณสมบัติสอดคล้องกับการจำแนกช่วงต้นได้ดี

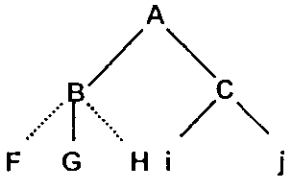
ผลการทดลองของงานวิจัยนี้ได้เลือกใช้ข้อมูลดีเอ็นเอของแบคทีเรียอี.โคไล แบ่งออกเป็น 2 กลุ่ม คือแบบโปรโมเตอร์และไม่เป็นโปรโมเตอร์ โดยใช้สายอักขระดีเอ็นเอซึ่งมีความยาว 57 ตัวอักษร จำนวนกลุ่มละ 53 สายรวมแล้ว 106 สายพบว่าข้อผิดพลาดในการจำแนกข้อมูลช่วงต้นด้วยวิธีการ SCR จำนวน 7 สายจาก 106 สาย โดยใช้ความยาวสายอักขระช่วงต้นความยาวเฉลี่ย 21/53 และวิธีการ GSDT มีความผิดพลาด 10 สายจาก 106 สาย โดยใช้ความยาวสายอักขระช่วงต้น 20/53

2.2.3. บทความเรื่อง Refinement of Approximate Domain Theories by Knowledge-Based Neural Networks (Towell, Shavlik, &Noordewier, 1990)

บทความได้กล่าวถึงการใช้วิธีการเรียนรู้แบบผสมผสาน ที่เรียกว่าโครงข่ายประสาทเทียมแบบใช้ฐานความรู้ (knowledge-based artificial neural networks: KBANN) โดยมีหลักการสำคัญคือ การเรียนรู้ข้อมูลเชิงลำดับนั้นในเริ่มต้นให้ทำการหากฎเกณฑ์ของลำดับที่เกิดขึ้นโดยใช้ตรรกศาสตร์ AND OR NOT เพื่อเชื่อมกฎเหล่านั้น โดยสมมติว่ากฎที่สร้างขึ้นนี้ไม่เป็นกฎเรียกซ้ำ (recursive) เช่น

กฎ A: B, C. **กฎ B: NOT** F, G. **กฎ B: NOT** H. **กฎ C:** I, J.

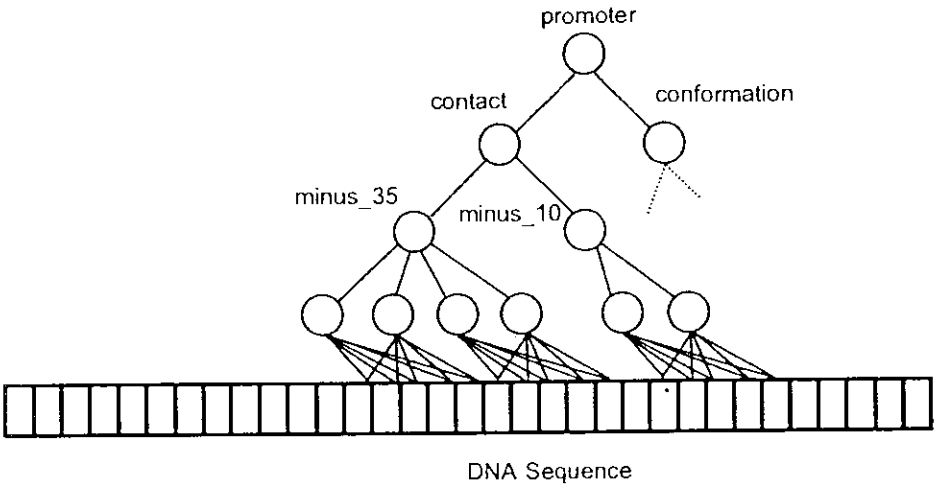
เมื่อได้กฎเหล่านี้มาแล้วจึงนำกฎเหล่านี้ไปแปลงเป็นส่วนหน่วย (units) และทำการเชื่อม (links) ในกระบวนการเริ่มต้นของโครงข่ายประสาทเทียม เช่นจากกฎที่กล่าวมาสามารถนำมาสร้าง หน่วยและเส้นเชื่อมดังนี้



ผู้วิจัยนำวิธีการ KBANN มาทดลองใช้กับการจำแนกสายอักขระดีเอ็นเอของแบคทีเรียอี.โคไลซึ่งเป็นสายอักขระของนิวคลีโอไทด์ (A, C, G, T) โดยแบ่งกลุ่มข้อมูลที่มีคุณสมบัติเป็นโปรโมเตอร์และไม่เป็นโปรโมเตอร์ อย่างละ 53 สาย รวมทั้งหมด 106 สาย แต่ละสายมีความยาว 57 ตัวอักษร โดยสำรวจกฎของความเป็นโปรโมเตอร์ซึ่งได้ตัวอย่างกฎเกณฑ์บางส่วนดังนี้ คือ ตัวที่เป็นโปรโมเตอร์จะประกอบไปด้วยสายอักขระ 2 ส่วนคือ ส่วนที่เป็น contact และ ส่วนที่เป็น conformation โดยส่วนที่เป็น contact ประกอบด้วยกฎ 2 ส่วนคือ minus_35 และ minus_10 โดยมีรายละเอียดของกฎดังนี้ ให้อักษร x นิวคลีโอไทด์ตัวใดๆก็ได้และ @-44 คือ ตำแหน่ง -44

```
Promoter:- contact, conformation.
contact:- minus_35, minus_10.
minus_35:- @-37 "cttgac".
minus_35:- @-36 "ttgxca".
minus_35:- @-36 "ttgaca".
minus_35:- @-36 "ttgac".
minus_10:- @-14 "tataat".
minus_10:- @-13 "taxxxt".
minus_10:- @-13  "tataat"
minus_10:- @-12 "taxxxt"
conformation:- @-45 "aaxxa"
conformation:- @-45 "axxxa", @-4 "t", @-28 "txxxtxaaxxtx".
conformation:- @-49 "axxxxt", @-1 "a", @-27 "txxxxaxxtxtg".
conformation:- @-47 "caaxttxac", @-22 "gxxxtxc", @-8 "gcgccxcc".
```

ซึ่งกฎดังกล่าวนี้จะสามารถนำมาสร้างเป็นโครงข่ายได้ดังรูป



รูปที่ **2.1** ภาพจำลองการสร้างโครงข่ายประสาทด้วยวิธีการ KBANN

หมายเหตุ.จาก "Refinement of Approximate Domain Theories by Knowledge-Based Neural Networks", โดย Towell, G. and Shavlik, 1990, In Proceedings of National Conference on Artificial Intelligence, 8, 861-866

การทดลองจากงานวิจัยนี้ใช้วิธีการทดสอบแบบดึงข้อมูลออกหนึ่งตัว (leave one out) ซึ่งข้อมูลที่ใช้ฝึกสอนมีจำนวนทั้งหมด 106 สายจะทำการดึงออก 1 สายแล้วใช้ที่เหลือฝึกสอน แล้วใช้ตัวที่ดึงออกนั้นเป็นตัวทดสอบเพื่อให้ตัวที่ดึงออกนั้นเป็นตัวที่แบบจำลองไม่เคยพบมาก่อน แล้วเปลี่ยนข้อมูลที่ดึงออกจนครบทั้ง 106 สาย ซึ่งวิธีการทดสอบนี้อาจจะค่อนข้างใช้เวลาสูงเนื่องจากต้องฝึกสอนถึง 106 ครั้ง ผลการทดลองพบว่า วิธีการ KBANN สามารถจำแนกข้อมูลได้ดี โดยมีความผิดพลาดเพียง 4 สายจาก 106 สาย ผู้วิจัยเทียบกับวิธีการอื่นได้แก่ วิธีการโครงข่ายประสาทเทียมแบบมาตรฐาน วิธีการเฉพาะทาง วิธีการต้นไม้ตัดสินใจ และวิธีการเพื่อนบ้านใกล้สุด ได้ผลการทดลองดังตาราง 2.1 ซึ่งแสดงให้เห็นว่าวิธีการ KBANN มีอัตราส่วนความผิดพลาดน้อยที่สุด

ตารางที่ 2.1 อัตราส่วนความผิดพลาดในการจำแนกข้อมูลในงานวิจัย

วิธีการจำแนก	อัตราส่วนความผิดพลาด
วิธีการ KBANN	4/106
วิธีการโครงข่ายประสาทเทียมมาตรฐาน	8/106
วิธีการเฉพาะของ O'neill	12/106
วิธีการเพื่อนบ้านใกล้สุด	13/106
วิธีการต้นไม้ตัดสินใจ	19/106

หมายเหตุ.จาก "Refinement of Approximate Domain Theories by Knowledge-Based Neural Networks", โดย Towell, G. and Shavlik, 1990, In Proceedings of National Conference on Artificial Intelligence, 8, 861-866

2.2.4. บทความเรื่อง: Minimal Unique Substring and Maximum repeat (Ilie & Smyth, 2011)

งานวิจัยนี้ได้นำเสนอและนิยามคำว่า สายอักขระเอกลักษณ์สั้นสุด และสายอักขระเกิดซ้ำยาวสุด ซึ่งเดิมทีนั้นทั้งสองคำนี้ได้ถูกกล่าวถึงในบทความอื่นๆประปราย มีนิยามไม่ค่อยชัดเจนเท่าไร แต่พบว่ามีความสำคัญเกี่ยวข้องกับปัญหาการจำแนกข้อมูลลำดับดีเอ็นเอ เนื่องจากข้อมูลดีเอ็นเอจะมีลักษณะเป็นสายอักขระ ผู้วิจัยได้นำเสนอทฤษฎีที่เกี่ยวข้องกับสายอักขระเอกลักษณ์สั้นสุด และสายอักขระเกิดซ้ำยาวสุด และขั้นตอนวิธีในการหาสายอักขระดังกล่าว

งานวิจัยก่อนหน้าที่ได้กล่าวถึงวิธีการหา สายอักขระเอกลักษณ์ที่สั้นที่สุดได้แก่ งานวิจัยซึ่งเสนอขั้นตอนวิธีหาต้นไม้คำเดิมท้าย ซึ่งเสนอโดย ไวเนอร์ (Weiner, 1973) โดยบทความดังกล่าวของไวเนอร์ ได้ให้นิยามสายอักขระเอกลักษณ์สั้นสุดของสายอักขระ w ใดๆคือ สายอักขระย่อย $w_{i..j}$ โดยที่ $w_{i..j-1}$ เป็นสายอักขระเกิดซ้ำอย่างน้อย 2 ครั้ง ดังนั้นการหาสายอักขระเกิดซ้ำ จึงสามารถทำได้โดยใช้ต้นไม้คำเดิมท้าย ซึ่งต่อมาได้พัฒนามาใช้ อาร์เรย์คำเดิมท้าย (suffix array) (Manber & Myers, 1990) ซึ่งประหยัดพื้นที่และเวลาในการดำเนินการมากกว่าต้นไม้คำเดิมท้าย โดยผู้วิจัยได้นำเสนอนิยามที่สำคัญ ซึ่งมีความสำคัญคือ สายอักขระเอกลักษณ์สั้นสุด และสายอักขระเกิดซ้ำยาวสุด ดังต่อไปนี้

สายอักขระเกิดซ้ำ (a repeat substring) ของ w คือ สายอักขระย่อยในตำแหน่ง i ถึง j แทนได้ด้วย $w[i..j]$ เกิดขึ้นอย่างน้อย 2 ครั้งในสายอักขระ w เมื่อมีการเกิดซ้ำแล้วสายอักขระย่อยภายในการเกิดซ้ำนั้นจะเกิดซ้ำเช่นเดียวกัน ดังนั้นแน่นอนว่าจะต้องมีสายอักขระเกิดซ้ำยาวสุด ซึ่งเป็นสายอักขระที่หากเพิ่มความยาวเลยไปทางซ้ายหรือทางขวาอีก 1 ตัวอักษรก็จะกลายเป็นสายอักขระไม่เกิดซ้ำหรือเรียกอีกอย่างคือ สายอักขระเอกลักษณ์ นิยามได้ว่า ให้ $w[i..j]$ เป็นสายอักขระเกิดซ้ำแล้ว และถ้าสายอักขระ $w[i-1..j]$และ $w[i..j+1]$ ไม่เป็นสายอักขระเกิดซ้ำแล้ว $w[i..j]$ เป็น*สายอักขระเกิดซ้ำยาวสุด* (maximum repeated substring)

สายอักขระเอกลักษณ์ (a unique substring) ของ w คือ สายอักขระย่อยในตำแหน่ง i ถึง j แทนได้ ด้วย $w[i..j]$ ที่เกิดขึ้นเพียงครั้งเดียวใน w โดยในการหาสายอักขระเกิดซ้ำนั้นหากพบสายอักขระเอกลักษณ์ตัวที่สั้นที่สุดแล้วก็ย่อมหมายความว่า สายอักขระย่อยที่ยาวกว่าและมีสายอักขระเอกลักษณ์สั้นสุดปนอยู่ในนั้น ก็ย่อมเป็นสายอักขระเอกลักษณ์ด้วย ดังนั้นสายอักขระเอกลักษณ์สั้นสุดย่อมเป็นตัวกำหนดสายอักขระเอกลักษณ์สายอื่นๆด้วย โดยนิยาม*สายอักขระเอกลักษณ์สั้นสุด* (minimum unique substring) $w[i..j]$ ใดๆทั้งในกรณี $i=j$ ซึ่งเป็นอักขระตัวเดียวได้เกิดขึ้นครั้งเดียว และ กรณี $i<j$ โดยที่ $w[i+1..j]$ และ $w[i..j-1]$ เป็นสายอักขระเกิดซ้ำยาวสุด

ตัวอย่าง $w=abaababa$ แล้ว aba ซึ่งเป็น $w[1..3], w[4..6], w[6..8]$ เป็นสายอักขระเกิดซ้ำยาวสุด และ $w[3..4]=aa$ และ $w[5..7]=bab$ เป็นสายอักขระเอกลักษณ์สั้นสุด

ผู้วิจัยสรุปเป็นทฤษฎีโดยมีใจความโดยย่อว่า *ทั้งสายอักขระเอกลักษณ์สั้นสุดและสายอักขระเกิดซ้ำยาวสุดนั้นจะเกิดควบคู่กันเสมอ* และเสนอขั้นตอนวิธีในการหาสายอักขระทั้งสองประเภทโดยใช้อาร์เรย์คำเดิมท้าย

2.2.5 บทความเรื่อง: Probabilistic Finite State machine part I (Vidal, Thollard, de la Higuera, Casacuberta, &Carrasco, 2005)

บทความสำรวจนี้ได้สรุปงานวิจัยเกี่ยวกับเครื่องจักรสถานะเชิงน่าจะเป็นแบบจำกัด (probabilistic finite state machine) ในเชิงทฤษฎี โดยนิยามภาษาและเครื่องจักรเชิงน่าจะเป็นแบบจำลองเชิงสถิติประเภทต่างๆ รวมถึงนิยามและคุณสมบัติที่สำคัญ เพื่อให้งานวิจัยมีความชัดเจนไปในทิศทางเดียวกัน โดยเน้นที่ออโตมาดาจำกัดเชิงน่าจะเป็น (probabilistic finite automata) เรียกโดยย่อว่า PFA

เครื่องจักรสถานะเชิงน่าจะเป็นแบบจำกัดนั้น ได้แก่ ออโตมาดาจำกัดเชิงน่าจะเป็น (probabilistic finite automata) แบบจำลองมาร์คอฟแบบซ่อน (hidden Markov model) หรือ

HMM ไวยากรณ์ปรกติสโทแคสติก (stochastic regular grammar) ลูกโซ่มาร์คอฟ (markov chain) เอ็นแกรม (n-gram) ต้นไม้คำเติมท้ายเชิงน่าจะเป็น (probabilistic suffix tree) และ ออโตมาตาจำกัดเชิงน่าจะเป็นแบบกำหนด (deterministic probabilistic automata) ซึ่งแบบจำลองทั้งหมดนี้เป็นแบบจำลองที่ผลิตการแจกแจง (distribution) บนเซตของสายอักขระ ลำดับคำ และ วลี โดยผู้เขียนเน้นทฤษฎีที่กล่าวถึง PFA เป็นหลัก ด้วยเหตุผลว่า

1. ทฤษฎีภาษารูปนัยเป็นพื้นฐานสำหรับนักวิจัยและวิศวกรในสาขาวิทยาการคอมพิวเตอร์ การเพิ่มส่วนของความน่าจะเป็นไปยังเครื่องจักรออโตมาตานั้นจึงเป็นเรื่องที่สร้างขึ้นบนฐานของประสบการณ์และการหยั่งรู้ของนักวิจัย ย่อมมีความเหมาะสมต่อการพัฒนาให้ดียิ่งขึ้น

2. มีการพิสูจน์ว่า PFA สามารถใช้แทนกลุ่มของการแจกแจงได้เทียบเท่ากับ HMM โดยใช้พื้นที่ที่เท่ากันและขั้นตอนวิธีที่ง่ายที่สุดได้พอกัน

3. PFA มีทั้งแบบกำหนด (deterministic) และแบบไม่กำหนด (non-deterministic) ซึ่งสามารถนำไปใช้งานได้ตรงกับความต้องการ และในการใช้งานจริงเชิงปฏิบัตินั้น PFA สามารถนำไปประยุกต์ใช้สร้าง แบบจำลองสถานะรูปแบบอื่นๆได้

สรุปนิยามที่สำคัญเกี่ยวกับ PFA ตามบทความมีดังนี้

2.2.5.1 ภาษาสโทแคสติก (stochastic languages)

ให้ Σ เป็น*เซตจำกัดของชุดตัวอักษร* และ Σ^* เป็น*เซตของสายอักขระที่เกิดจาก* Σ และ *สายอักขระว่าง*แทนได้ด้วย λ [1]

ภาษา (a language) หมายถึงเซตย่อยของ Σ^* และความยาวของสายอักขระ $x \in \Sigma^*$ แทนด้วย $|x|$ เซตของสายอักขระความยาว n ความยาวน้อยกว่า n และความยาวน้อยกว่าหรือเท่ากันกับ n แทนได้ด้วยด้วย Σ^n, $\Sigma^{<n}$, $\Sigma^{\leq n}$ ตามลำดับ สายอักขระย่อยของ x จากตำแหน่ง i ถึงตำแหน่ง j แทนด้วย $x_i \ldots x_j$ และในกรณีที่ $j < i$ แล้ว $x_i \ldots x_j = \lambda$

ภาษาสโทแคสติก (a stochastic language) D คือ การแจกแจงความน่าจะเป็นบน Σ^* โดยแทนความน่าจะเป็นของสายอักขระ x ด้วย $Pr_D(x)$ โดย $x \in \Sigma^*$และการแจกแจงนี้จะต้องระบุชัดว่า

$$\sum_{x \in \Sigma^*} Pr_D(x) = 1$$

---

[1] สายอักขระว่าง (empty string) อาจใช้แทนได้ด้วย λ เพื่อให้คงไว้ตามบทความเดิม แต่จะใช้สัญลักษณ์ ε ในงานวิจัยนี้

หากการแจกแจงนี้สามารถถูกแทนได้ด้วยแบบจำลอง A จะแทนได้ด้วย $Pr_A(x)$ และภาษาสโทแคสติก D ที่ถูกจำลองขึ้นจากเครื่องจักร A แทนได้ด้วย D_A และหากว่า L เป็นภาษาหนึ่งบน Σ และ D เป็นการแจกแจงบน Σ^* แล้ว

$$Pr_D(L) = \sum_{x \in L} Pr_D(x)$$

ชุดตัวอย่าง (a sample) S คือ เซตของสายอักขระหลายเซต โดยสายอักขระ $x \in S$ หมายถึงสายอักขระ x อาจปรากฏใน S ซึ่งอาจจะปรากฏครั้งเดียวหรือหลายครั้งก็ได้ ขนาดของ S แทนด้วย $|S|$ หมายถึงจำนวนสายอักขระใน S และ $\|S\|$ หมายถึงความยาวรวมของสายอักขระทุกสายใน S การแจกแจงอย่างง่ายซึ่งใช้ร่วมกับ S แทนด้วย D_S คือ

$$Pr_{D_S}(x) = f(x)/|S|$$

โดยให้ $f(x)$ คือ ความถี่ของการเกิด x ใน S และ $Pr_{D_S}(X) = 0$ ในกรณีที่ $x \notin S$

2.2.5.2 ออโตมาตาเชิงน่าจะเป็น (probabilistic finite automata)

ออโตมาตาเชิงน่าจะเป็นสามารถนิยามได้อย่างเป็นรูปนัยดังนี้ กำหนดให้ PFA $A = [Q_A, \Sigma, \delta_A, I_A, P_A, F_A]$ โดยที่

Q_A คือ เซตจำกัดของสถานะ(a finite set of states)

Σ คือ ชุดตัวอักษร (a finite set of alphabets)

$\delta_A \subseteq Q \times \Sigma \times Q_A$ คือ เซตของการเปลี่ยนสถานะ (a set of transitions)

$I_A : Q_A \to \mathbb{R}^+$ คือฟังก์ชันความน่าจะเป็นของสถานะเริ่มต้น (the initial probability function)

$P_A : \delta_A \to \mathbb{R}^+$ คือฟังก์ชันความน่าจะเป็นของการเปลี่ยนสถานะ (the transitions probability function)

$F_A : Q_A \to \mathbb{R}^+$ คือ ความน่าจะเป็นของการสถานะสุดท้าย (the final state probability function)

โดยกำหนดให้ I_A, P_A, F_A เป็นฟังก์ชันที่มีคุณสมบัติคือ

$$\sum_{q \in Q_A} I_A(q) = 1$$

และ

$$\forall q \in Q_A,\ F_A(q) + \sum_{a \in \sum,\, q' \in Q_A} P_A(q, a, q') = 1$$

2.2.5.3 ออโตมาตาเชิงน่าจะเป็นแบบ λ (λ probabilistic automata)

ออโตมาตาเชิงน่าจะเป็นแบบ λ มีนิยามเหมือนกันกับ PFA แต่มีส่วนของเซตการเปลี่ยนสถานะนิยามได้ด้วย $\delta \subseteq Q \times ((\Sigma \cup \{\lambda\}) \times Q)$ ซึ่งใช้สำหรับในกรณีที่การเปลี่ยนสถานะด้วย λ

2.2.5.4 ออโตมาตาเชิงน่าจะเป็นแบบกำหนด (deterministic probabilistic finite state automata)

ออโตมาตาเชิงน่าจะเป็นแบบกำหนด เรียกโดยย่อว่า DPFA มีข้อได้เปรียบคือ

1. การเปลี่ยนสถานะดำเนินการได้ในเส้นทางเดียว ซึ่งง่ายต่อการดำเนินการ
2. ปัญหาที่ไม่สามารถจัดการได้ (intractable problems) (มีขั้นตอนวิธีที่มีอัตราการเดิบโตของเวลาประมวลผลสูง) เช่น ปัญหาสายอักขระที่มีความน่าจะเป็นสูงสุด จะเปลี่ยนมาเป็นปัญหาที่สามารถจัดการได้
3. ปัญหาในการเรียนรู้บางอย่างสามารถดำเนินการได้ใน DPFA แต่ไม่ได้ใน PFA

DPFA มีนิยามดังนี้คือ

ให้ DPFA $A = [Q, \Sigma, \delta, I, P, F]$ โดยที่มีเงื่อนไขคือ

$\exists q_0 \in Q$ โดยที่ $I(q_0) = 1$

$\forall \in Q, \forall \in \Sigma, |\{q' : (q, a, q') \in \delta\}| \leq 1$

2.2.5.5 ขนาดของ PFA (size of PFA)

การสร้าง PFA จะพิจารณา 2 ประเด็นคือ พิจารณาจากจำนวน Q และความซับซ้อนของขั้นตอนวิธีในการสร้าง PFA โดยความซับซ้อนนั้นจะต้องอยู่ในกำหนดเวลาเชิงพหุนามเมื่อเทียบกับจำนวนหน่วยที่นำไปใช้ในการสร้าง PFA จึงจะยอมรับได้ว่าเป็นปัญหาที่จัดการได้

2.2.5.6 การจำลองการแจกแจงด้วยออโตมาตาเชิงน่าจะเป็น (distribution modeled by PFA)

ให้เส้นทาง (path) $\theta = (s_0, x_1, s_1, x_2, s_2, \ldots, s_{k-1}, x_k, s_k)$ สำหรับสายอักขระ x ที่เกิดขึ้นใน A และ s_i คือสถานะลำดับที่ i ซึ่งจะมีลำดับของการเปลี่ยนสถานะได้ แก่ $(s_0, x_1, s_1), (s_1, x_2, s_2), \ldots, (s_{k-1}, x_k, s_k)$ โดยมี $x = x_1 x_2 \ldots x_k$ แล้ว ความน่าจะเป็นในการสร้างเส้นทาง คือ

$$Pr_A(\theta) = I(s_0) \cdot (\prod_{j=1}^{k} P(s_{j-1}, x_j, s_j)) \cdot F(s_k)$$

เส้นทางที่ใช้การได้ (a valid path) คือเส้นทางสำหรับสายอักขระ x ใดๆซึ่งมีความน่าจะเป็นมากกว่า 0 และให้เซตของเส้นทางที่ใช้การได้ทั้งหมดของ A แทนด้วย Θ_A และเซตของเส้นทางใช้การได้ทั้งหมดสำหรับสายอักขระ x ของ A แทนด้วย $\Theta_A(x)$ โดยหากว่า $|\Theta_A(x)| > 1$ จะถือว่า PFA นั้นมีความกำกวม (ambiguous) และหากในกรณีที่ PFA มีความกำกวมแล้ว ความน่าจะเป็นในการสร้าง x ด้วย A จะคำนวณได้จาก

$$Pr_A(x) = \sum_{\theta \in \Theta_A(x)} Pr_A(\theta) \qquad (1)$$

อย่างไรก็ตาม DPFA จะต้องไม่เกิดความกำกวม

2.2.5.7 ความต้องกันของออโตมาตาเชิงน่าจะเป็น (consistency of PFA)

ความต้องกันของออโตมาตาเชิงน่าจะเป็น หมายถึง คุณสมบัติที่ว่าผลรวมความน่าจะเป็นในทุกเส้นทางที่เป็นไปได้จะต้องเท่ากับ 1 และสถานะทุกสถานะต้องมีประโยชน์ กล่าวคือ มีเส้นทางใช้การได้อย่างน้อยหนึ่งเส้นทาง ซึ่งหากมีสถานะใดที่ไม่มีสายอักขระไปถึงได้ด้วยความน่าจะเป็นมากกว่า 0 แล้ว สถานะนั้นไร้ประโยชน์

2.2.5.8 การแจงส่วน (parsing issues)

การแจงส่วนในที่นี้คือ การคำนวณความน่าจะเป็นของสายอักขระ ซึ่งในกรณี DPFA จะสามารถคำนวณได้ง่ายกว่า PFA เนื่องจากการคำนวณจะใช้เส้นทางคำนวณเส้นทางเดียว แต่

PFA อาจยอมให้มีเส้นทางได้หลายเส้นทาง โดยนิยามการคำนวณความน่าจะเป็นของสายอักขระ $x_1x_2\ldots x_i$ จนเข้าถึงสถานะ q ในกรณีหลายเส้นทางด้วยฟังก์ชัน $\alpha_x(i,q)\ \forall_q \in Q$ และ $0 \le i \le |x|$ โดยที่

$$\alpha_x(i,q) = \sum_{(s_0,s_1,\ldots,s_i)\in\Theta_A(x_1\ldots x_i)} I(s_0)\cdot\prod_{j=1}^{i} P(s_{j-1},x_j,s_j)\cdot 1(q,s_i) \qquad (2)$$

เมื่อ $1(q,q')=1$ เมื่อ $q=q'$ และ $1(q,q')=0$ เมื่อ $q\neq q'$

2.2.5.9 การหาเส้นทางสำหรับสายอักขระที่ดีที่สุดใน PFA (optimal path for a string)

การหาความน่าจะเป็นของการผลิตสายอักขระ x ด้วย A นั้นสามารถนิยามได้จากผลรวมของความน่าจะเป็นจากทุกเส้นทางที่เป็นไปได้ แต่จะมีเส้นทางหนึ่งที่ให้ค่าความน่าจะเป็นดีที่สุด ซึ่งสามารถนิยามได้ดังนี้คือ

$$\tilde{\theta} = \arg\max_{\theta\in\Theta_A(x)} Pr_A(\theta)$$

ความน่าจะเป็นที่ได้จากเส้นทาง $\tilde{\theta}$ สามารถแทนได้ด้วย $\tilde{Pr}_A(x)$ ซึ่งหาได้จากฟังก์ชัน $\gamma_x(i,q)\ \forall q\in Q, 0\le i\le |x|$ ซึ่งเป็นฟังก์ชันหาความน่าจะเป็นสูงสุดในการผลิตสายอักขระตำแหน่งหน้า $x_1\ldots x_i$

$$\gamma_x(i,q) = \max_{(s_0,s_1,\ldots,s_i)\in\Theta_A(x_1\ldots x_i)} I(s_0)\cdot\prod_{j=1}^{i} P(s_{j-1},x_j,s_j)\cdot 1(q,s_i)$$

โดยที่

$$\tilde{Pr}_A = \max_{q\in Q}\gamma_x(|x|,q)\cdot F(q)$$

2.2.5.10 ปัญหาความน่าจะเป็นมากที่สุดของสายอักขระ

เป็นปัญหาในการค้นหาว่า สายอักขระใดใน D_A ที่มีความน่าจะเป็นสูงสุด โดยนิยามด้วยสมการ

$$\arg\max_{x\in\Sigma^*} Pr_A(x)$$

2.2.5.11 การเปรียบเทียบการแจกแจง

การวัดความคล้ายกันของการแจกแจงถูกใช้ในการเรียนรู้ ด้วยการวัดว่าข้อมูลตัวอย่างที่ใช้สำหรับการเรียนรู้ PFA นั้นเมื่อนำไปทดลองกับข้อมูลทดสอบแล้วแตกต่างจากแบบจำลองอีกตัวหนึ่งอย่างไร ทั้งนี้เพื่อตรวจสอบว่าแบบจำลองที่ปรับเปลี่ยนพารามิเตอร์แล้วนั้นจะให้ผลแตกต่างกับตัวที่นับจำนวนแบบตรงไปตรงมาหรือไม่ โดยสามารถใช้การวัด 2 ประเภทได้แก่ ระยะทางเชิงคณิตศาสตร์ และการวัดด้วยพื้นฐานเอนโทรปี

การวัดระยะทางเชิงคณิตศาสตร์

สามารถทำได้โดยให้ d_n แทนระยะทางสำหรับภาษา L_n

$$d_n(D, D') = \left(\sum_{x\in\Sigma^*} |Pr_D(x) - Pr_{D'}(x)|^n \right)^{\frac{1}{n}}$$

หรือใช้ระยะทางเชิงลอการิทึม

$$d_{\log}(D, D') = \max_{x\in\Sigma^*} |\log Pr_D(x) - \log Pr_{D'}(x)|$$

การวัดด้วยพื้นฐานเอนโทรปี

สามารถทำได้โดยใช้ไดเวอร์เจนซ์ของคุลแบ็ค-ลีบเลอร์ (Kullback-Leibler divergence) โดยมีสมการดังนี้

$$d_{KL}(D, D') = \sum_{x\in\Sigma^*} (Pr_D(x) \cdot \log Pr_D(x) - Pr_D \cdot \log Pr_{D'}(x))$$

2.2.6 บทความเรื่อง: Probabilistic finite state machine part II (Vidal, Thollard, de la Higuera, Casacuberta, &Carrasco, 2005)

ในบทความสำรวจส่วนที่ 2 นี้ได้สรุปเนื้อหาเครื่องจักรสถานะจำกัดประเภทอื่นนอกจากพีเอฟเอ(PFA) เช่น เอ็นแกรม และมาร์คอฟแบบซ่อน โดยสรุปขั้นตอนวิธีในการคำนวณความน่าจะเป็น นอกจากนี้แล้วยังมีประเด็นที่น่าสนใจเกี่ยวกับการใช้งานพีเอฟเอ เช่น ขั้นตอนวิธีการเรียนรู้ และกระบวนทัศน์การเรียนรู้ โดยมีสาระสำคัญดังต่อไปนี้

2.2.6.1 เครื่องจักรสถานะอื่น ๆที่น่าสนใจ

เนื่องจากมีเครื่องจักรสถานะต่าง ๆเกิดขึ้นมากมาย ซึ่งถูกนำไปใช้ในการคำนวณหาความน่าจะเป็นของสายอักขระ ประเด็นที่น่าสนใจคือ ความแตกต่างของการนิยามการคำนวณความน่าจะเป็นของเครื่องจักรแต่ละประเภทนั้นอยู่ตรงจุดไหน แม้ว่าโดยส่วนใหญ่แล้วแบบจำลองประเภทนี้จะใช้หลักการคำนวณบนสายอักขระความยาวจำกัดหรือ Σ^n แต่ค่า n นั้นแตกต่างกันไป ทำให้ไม่สามารถนำเอาความน่าจะเป็นมาเปรียบเทียบกับสายอักขระที่มีความยาวแตกต่างกันได้ อย่างไรก็ตามเครื่องจักรสถานะบางประเภทถูกนำมาใช้คำนวณความน่าจะเป็นของสายอักขระความยาวไม่จำกัด Σ^* ได้ เช่น มาร์คอฟแบบซ่อนและเอ็นแกรม โดยมีรายละเอียดดังนี้

2.2.6.2 เอ็นแกรม (N-gram)

เอ็นแกรมเป็นแบบจำลองที่ถูกใช้อย่างกว้างขวางเกี่ยวกับการประมวลผลภาษาธรรมชาติ (natural language processing) (Cavnar & Trenkle, 1994) การจดจำเสียง (speech recognition) การจดจำลายมือ (handwriting text recognition) เป็นต้น โดยแบบจำลองเอ็นแกรมนั้นจะใช้คำนวณหาการแจกแจงของสายอักขระความยาวคงที่ โดยให้สายอักขระ x ใดๆมีความยาว m แล้ว สามารถคำนวณหาความน่าจะเป็นตามกฎลูกโซ่ของมาร์คอฟได้ดังต่อไปนี้

$$Pr(x) = Pr(x_1) \cdot \prod_{l=2}^{m} Pr(x_l | x_1, \ldots, x_{l-1})$$

ในกรณีที่สายอักขระความยาวมากจะประมาณความน่าจะเป็นในการเกิดอักขระใดจากสายอักขระก่อนหน้า $n-1$ ตัวอักษร ได้จาก

$$Pr(x) \approx \prod_{l=1}^{m} Pr(x_l | x_{l-n+1}, \ldots, x_{l-1})$$

โดยหาก n มีค่าเท่ากับ 2 เรียกว่าไบแกรม (bigrams) ซึ่งการเรียนรู้แบบจำลองเอ็นแกรมทำได้ด้วยการอ่านชุดข้อมูลเรียนรู้แล้วใช้การนับความถี่ในการเกิดสายอักขระ โดยให้ S เป็นข้อมูลที่ใช้เรียนรู้แล้ว

$$P_n(a | z) = f(za)/f(z)$$

และ $a \in \Sigma \cup \{\#\}, z \in \Sigma^{\leq n}$ และ $f(y)$ คือความถี่ของการเกิด y ใน S

2.2.6.3 แบบจำลองมาร์คอฟแบบซ่อน (Hidden Markov Model)

แบบจำลองมาร์คอฟแบบซ่อนถูกนำมาใช้ในงานเช่นเดียวกันกับเอ็นแกรม นอกจากนี้แล้วยังนำมาใช้ในเรื่องเกี่ยวกับการจดจำรูปแบบ (pattern recognition) ได้เป็นอย่างดี เช่น การจดจำรูปร่างวัตถุ การจดจำภาพ การวิเคราะห์ภาพเชิงการแพทย์ ดูรายละเอียด (Rabiner, 1989) โดยแบบจำลองมาร์คอฟแบบซ่อน (HMM) มีนิยามดังต่อไปนี้

HMM ประกอบไปด้วย 6 องค์ประกอบ คือ $\mathfrak{M} = \langle Q, \Sigma, I, q_f, T, E \rangle$ โดยที่

Q เป็นเซตจำกัดของสถานะ

Σ เป็นชุดตัวอักษรของสัญลักษณ์

$I : Q - \{q_f\} \rightarrow \mathbb{R}^+$ เป็นฟังก์ชันความน่าจะเป็นเริ่มต้น

$q_f \in Q$ เป็นสถานะยอมรับ

$T : (Q - \{q_f\} \times Q \rightarrow \mathbb{R}^+$ เป็นฟังก์ชันสถานะไปยังสถานะเชิงน่าจะเป็น

$E : (Q - \{q_f\} \times \Sigma \rightarrow \mathbb{R}^+$ เป็นฟังก์ชันเปลี่ยนสถานะเชิงน่าจะเป็น

โดยมีเงื่อนไขคือ

$$\sum_{q \in Q - \{q_f\}} I(q) = 1,$$

$$\sum_{q' \in Q} T(q, q') = 1, \forall q \in Q - \{q_f\}$$

$$\sum_{a \in \Sigma} E(q, a) = 1, \forall q \in Q - \{q_f\}$$

แบบจำลอง $\mathfrak{M}$ สามารถคำนวณความน่าจะเป็นของสายอักขระ $x = x_1 \ldots x_k$ ด้วย $Pr_{\mathfrak{M}}(x)$ โดยให้ θ เป็นเส้นทางซึ่งมีความยาว k แล้ว ให้ลำดับของสถานะได้แก่ $(s_1, s_2, \ldots, s_k)$ แล้วความน่าจะเป็นของ θ คือ

$$Pr_{\mathfrak{M}}(\theta) = I(s_1) \cdot \prod_{2 \le j \le k} T(s_{j-1}, s_j)$$

และความน่าจะเป็นของสายอักขระ x ตามเส้นทาง θ คือ

$$Pr_{\mathfrak{M}}(x \mid \theta) = \prod_{1 \le j < k} E(s_j, x_j)$$

และให้ $\Theta_{\mathfrak{M}}(x)$ เป็นเซตของเส้นทางทุกเส้นทางที่เป็นไปได้สำหรับ x แล้ว $Pr_{\mathfrak{M}}(x)$ สามารถคำนวณได้จาก

$$Pr_{\mathfrak{M}}(x) = \sum_{\theta \in \Theta_{\mathfrak{M}}(x)} Pr_{\mathfrak{M}}(x \mid \theta) \cdot Pr_{\mathfrak{M}}(\theta)$$

2.2.6.4 การเรียนรู้ออโตมาตาเชิงน่าจะเป็น (Learning Probabilistic Automata)

การเรียนรู้ PFA คือ การสร้างโครงสร้าง (สถานะและเส้นเชื่อมการเปลี่ยนสถานะ) และการประมาณพารามิเตอร์ต่างๆในองค์ประกอบของ PFA จากชุดข้อมูล เพื่อนำ PFA ที่มีองค์ประกอบต่างๆครบจากการเรียนรู้นั้น มาใช้งานเพื่อการทำนายข้อมูล เพื่อจดจำข้อมูล และการนำไปใช้ทางชีววิทยา เช่น การทำนายโครงสร้างของโปรตีน เป็นต้น โดยมีเป้าหมายว่าข้อมูลต่างๆนั้นสมมติว่าถูกสร้างขึ้นมาจากออโตมาตาตัวที่เรียนรู้นี้ ดังนั้น PFA จะต้องทำนายหรือ จดจำ รูปแบบข้อมูลได้ถูกต้องตามความเป็นจริง โดยการเรียนรู้นั้นแบ่งได้เป็น 2 แบบคือ 1) ใช้การประมาณความน่าจะเป็นจากโครงสร้างของ PFA ที่มีอยู่ก่อน และ 2) การหาโครงสร้างและประมาณความน่าจะเป็นของโครงสร้างนั้นไปพร้อมๆกัน

1) การประมาณความน่าจะเป็นของ PFA

การประมาณความน่าจะเป็นของ PFA นั้นตั้งอยู่บนสมมติฐานว่า โครงสร้างของ PFA นั้นมีอยู่ก่อนหน้า เช่น โครงสร้างของ PFA ที่ใช้กับแบบจำลองเอ็นแกรม จะมีขนาดโครงสร้างเทียบกับค่าของ n แกรม ดังนั้นวิธีการคือ ประมาณค่าพารามิเตอร์อันได้แก่ I P F ของ PFA ที่

ทำให้ความน่าจะเป็นในการคำนวณค่า x ใดๆจากอินพุตให้มีค่าสูงที่สุด โดยใช้หลักการ Maximum Likelihood (ML) ดังนี้

$$(\hat{I}, \hat{P}, \hat{F}) = \arg\max_{I,P,F} \prod_{x \in S} Pr_A(x)$$

การประมาณค่า I, P, F ทำได้โดย ใช้วิธี การนับจำนวนความถี่ของการเปลี่ยนสถานะเมื่อรับข้อมูลจาก S เข้ามา ซึ่งทำได้ง่ายในกรณีที่ PFA เป็นแบบเชิงกำหนด อย่างไรก็ตามในกรณีที่ PFA เป็นแบบในเชิงไม่กำหนดแล้วจะทำได้ค่อนข้างยาก ปัจจุบันมีขั้นตอนวิธีที่ใช้สำหรับกรณี PFA เชิงไม่กำหนดคือ ขั้นตอนวิธี Baum-Welch

2) การเรียนรู้โครงสร้างของ PFA

การเรียนรู้โครงสร้างนั้นจำเป็นจะต้องคำนึงถึง กระบวนทัศน์ในการเรียนรู้ (learning paradigm) ซึ่งหมายถึง มุมมองประเด็นการนิยามความสำเร็จของการเรียนรู้ เช่น การได้โครงสร้างมาจากการเรียนรู้นั้นจะสำเร็จเมื่อไร ใช้อะไรเป็นเกณฑ์วัดความสำเร็จ ซึ่งถูกเสนอครั้งแรกโดย Gold (Gold, 1967) มีชื่อว่า การเรียนรู้แบบจำแนกได้ภายในขอบเขตจำกัดด้วยความน่าจะเป็นรวมเท่ากับ 1 กล่าวโดยย่อคือ *หากมีข้อมูลตัวอย่างสำหรับการเรียนรู้ไม่จำกัดจำนวน ซึ่งถูกแจกแจงมาจากเป้าหมายที่เราไม่ทราบแล้ว ขั้นตอนวิธีการเรียนรู้จะต้องคืนค่าสมมติฐานเมื่อได้รับข้อมูลตัวอย่างใหม่เข้ามา โดยกล่าวว่า กลุ่มข้อมูลที่เรียนรู้นั้นสามารถจำแนกได้ภายในขอบเขตจำกัดด้วยความน่าจะเป็นรวมเท่ากับ 1 ถ้าหากขั้นตอนวิธีนั้นสามารถแสดงเป้าหมายจริงได้โดยสมมติฐานที่ได้จากการเรียนรู้นั้นสมมูลกันด้านใดด้านหนึ่งกับเป้าหมาย* กล่าวอย่างง่ายคือ การเรียนรู้นี้จะยอมให้สร้างสมมติฐานใหม่จนกว่าจะพบตัวที่สอดคล้องกับเป้าหมายจริง โดยจะต้องมีหลักการว่าสมมติฐานใหม่นั้นจะสามารถยุติได้เมื่อเจอคำตอบ

กระบวนทัศน์ในการเรียนรู้แบบนี้มีข้อด้อยคือ

1. กระบวนทัศน์นี้ไม่ได้กล่าวถึงข้อจำกัดของความซับซ้อน(complexity) ของขั้นตอนวิธีการเรียนรู้
2. ไม่สามารถทราบได้ว่าข้อมูลจำนวนเท่าไรจึงจะเพียงพอต่อการเรียนรู้
3. แม้ว่าขั้นตอนวิธีสามารถพิสูจน์ได้ว่าสามารถจำแนกได้ภายในขอบเขตจำกัด แต่ก็อาจจะคืนคำตอบที่ผิดได้

แต่ถึงกระนั้นกระบวนทัศน์นี้ก็ยังถือเป็นเงื่อนไขสำคัญในการเรียนรู้ โดยหากพบว่าไม่สามารถพบเป้าหมายภายใต้เงื่อนไขนี้แล้วแสดงว่า เป้าหมายการเรียนรู้นั้นไม่สามารถเรียนรู้ได้

กระบวนทัศน์อีกแบบหนึ่งเสนอโดย Valiant (Valiant, 1984) เรียกว่า การเรียนรู้แบบประมาณความถูกต้องอย่างเป็นไปได้ (probably approximately correct :PAC) ซึ่งมีข้อบังคับที่ว่า ตัวผู้เรียน จะต้องคืนค่าเป้าหมายที่ถูกประมาณการว่าดีด้วยความน่าจะเป็นที่มีค่าสูง ซึ่งยืดหยุ่นมากกว่าการเรียนรู้โครงสร้างแบบของ Gold

2.3 สรุป

บทความวิจัยที่ได้สรุปมาข้างต้นนั้น มีประเด็นที่น่าสนใจ 4 ประเด็นคือ 1) วิธีการเรียนรู้เพื่อจำแนกข้อมูลนั้นแบ่งออกเป็น 2 แบบคือ เรียนรู้จากข้อมูลเชิงตัวเลข และเรียนรู้จากข้อมูลเชิงสัญลักษณ์ โดยการเรียนรู้จากข้อมูลเชิงตัวเลขนั้นมีข้อเสียเปรียบคือ ไม่สามารถนำผลการเรียนรู้นั้นไปแปลความหมายเป็นอย่างอื่น และใช้เวลาช้าในกรณีที่เวกเตอร์ของลักษณะเฉพาะนั้นมีขนาดใหญ่ แต่การเรียนรู้จากข้อมูลเชิงสัญลักษณ์นั้นยังขาดขั้นตอนวิธีการเรียนรู้ที่สามารถจำแนกข้อมูลที่แม่นยำ 2) ประเด็นปัญหาในการเรียนรู้ส่วนเพิ่มและการตอบสนองในการตัดสินใจจำแนกข้อมูลได้อย่างรวดเร็วนั้นเป็นปัญหาที่ท้าทายและต้องการแบบจำลองที่รองรับ 3) ปัญหาทางชีวสารสนเทศนั้นยังต้องการขั้นตอนวิธีในการเรียนรู้ข้อมูลรวมถึงการนำผลไปวิเคราะห์แก้ปัญหาอื่นๆจากข้อมูลด้วย โดยสายอักขระเอกลักษณ์สั้นสุดและสายอักขระเกิดซ้ำยาวสุด อาจจะเป็นปัจจัยที่ช่วยทำให้เกิดขั้นตอนวิธีที่ดีขึ้นได้ 4) การเรียนรู้ของเครื่องจักรสถานะเชิงน่าจะเป็นเช่น PFA มีศักยภาพที่ดีในการจำแนกสายอักขระ มีทฤษฎีรองรับและสามารถปรับปรุงเพื่อการเรียนรู้ได้ โดยพิสูจน์แล้วว่าเทียบเท่ากับแบบจำลองมาร์คอฟแบบซ่อน จากประเด็นดังกล่าวนี้นำไปสู่งานวิจัยนี้เพื่อค้นคว้าหาแบบจำลองการเรียนรู้ข้อมูลลำดับเชิงสัญลักษณ์ที่สามารถรองรับความต้องการได้หลากหลายยิ่งขึ้น โดยได้นำเสนอในบทถัดไป

บทที่ 3
วิธีการวิจัยและทฤษฎีที่นำเสนอ

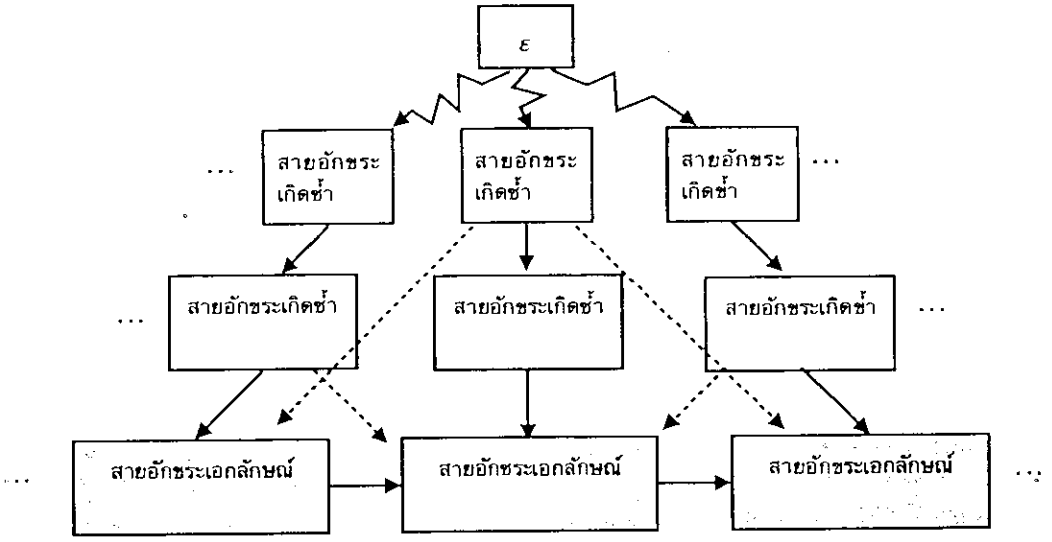
3.1 บทนำ

การจัดเก็บข้อมูลตัวอย่างสายอักขระที่แตกต่างนั้นสามารถนำไปใช้เพื่อค้นหาคำที่แตกต่างกันไป เช่น การหาคำนำหน้าร่วม การหาสายอักขระย่อย การหาคำเติมท้าย เป็นต้น การหาคำเหล่านี้จากข้อมูลขนาดใหญ่มักจะใช้ ต้นไม้คำเติมท้ายเพื่อเก็บตัวอย่างสายอักขระที่แตกต่างๆที่ได้จากข้อมูล ซึ่งตัดคำที่มีความยาวไม่จำกัดด้วยการใช้คำนำหน้าร่วม (common prefix) ทำให้โครงสร้างของต้นไม้ไม่ลึกเกินไปอย่างไม่จำกัด อย่างไรก็ตามในการใช้งานเพื่อวิเคราะห์ข้อมูล นอกจากการค้นหาคำแล้วบางครั้งอาจมีความต้องการวิเคราะห์ว่า สายอักขระย่อยที่ต้องการตรวจสอบนั้นเกิดขึ้นในข้อมูลหรือไม่หรือเกิดขึ้นจำนวนกี่ครั้งและมีความน่าจะเป็นในการเกิดสายอักขระต่างๆนั้นเท่าไร เพื่อใช้สำหรับการจำแนกกลุ่มของข้อมูลหรือเพื่อทำนายข้อมูลที่จะเกิดขึ้นในแต่ละขณะ การใช้ต้นไม้คำเติมท้ายสำหรับวิเคราะห์และคำนวณความน่าจะเป็นของสายอักขระย่อยนั้นมีข้อจำกัด ไม่สามารถหาความน่าจะเป็นของสายอักขระในระดับแต่ละตัวอักษรได้ดังเช่นแบบจำลองภาษาเชิงน่าจะเป็นเช่น ออโตมาตาเชิงน่าจะเป็น อีกทั้งด้วยรูปแบบของต้นไม้คำเติมท้ายนั้นเป็นโครงสร้างที่เก็บข้อมูลซึ่งแตกต่างจากแบบจำลองภาษาเชิงน่าจะเป็นที่เป็นโครงสร้างสำหรับวิเคราะห์ข้อมูล

ด้วยการพิจารณาข้อดีของโครงสร้างทั้ง 2 แบบ ในงานวิจัยนี้จึงเสนอแบบจำลองแบบใหม่ขึ้น ซึ่งเป็นแบบจำลองสถานะที่สามารถเก็บตัวอย่างสายอักขระที่แตกต่างได้คล้ายกับต้นไม้คำสำหรับใช้ในการตรวจสอบหาสายอักขระย่อยที่ปรากฏขึ้นในข้อมูล โดยใช้หลักการตัดความยาวของสายอักขระด้วยสายอักขระย่อยเอกลักษณ์ที่สั้นที่สุด ซึ่งคล้ายกับต้นไม้คำเติมท้ายที่สามารถตัดคำเติมท้ายเมื่อพบคำนำหน้าร่วม โดยแบบจำลองจะมีโครงสร้างคล้ายกับโครงสร้างต้นไม้ โดยมีสถานะแต่ละสถานะเก็บข้อมูลสายอักขระเกิดซ้ำ แต่มีความลึกไปถึงสถานะที่เก็บสายอักขระเอกลักษณ์ซึ่งเชื่อมโยงถึงกันแบบออโตมาตา โดยมีลักษณะดังรูปที่ 3.1

ด้วยลักษณะที่ออกแบบนี้ทำให้แบบจำลองนี้สามารถนำผลการเรียนรู้ไปวิเคราะห์รูปแบบเหตุการณ์อื่นๆนอกเหนือจากเพียงแค่การจำแนกข้อมูลได้ รวมถึงแบบจำลองนี้สามารถใช้วิเคราะห์คำนวณความน่าจะเป็นของสายอักขระใดๆในระดับแต่ละตัวอักษรได้เช่นเดียวกันกับออโตมาตาเชิงน่าจะเป็น ซึ่งจะทำให้เกิดข้อดีคือ การคำนวณความน่าจะเป็นของแบบจำลองในงานวิจัยนี้มีความแม่นยำมากยิ่งขึ้น เพราะสามารถหาความสัมพันธ์ของสายอักขระที่ขึ้นอยู่กับระยะยาวได้ตามสายอักขระย่อยเอกลักษณ์ที่สั้นที่สุด รวมถึงคำนำหน้าของสายอักขระย่อยเอกลักษณ์นั้น ซึ่งทำให้คำนวณความน่าจะเป็นของสายอักขระเกิดซ้ำได้ถึงความยาวของสายอักขระย่อย

เอกลักษณ์ที่สั้นที่สุด ซึ่งตามหลักการแล้วจะไม่มีความน่าจะเป็นใดมีค่าสูงกว่าการเกิดสายอักขระเอกลักษณ์เนื่องจากเกิดขึ้นเพียงครั้งเดียว ความน่าจะเป็นของข้อมูลที่ใกล้เคียงกับสายอักขระเอกลักษณ์ย่อมให้ค่าความน่าจะเป็นสูงสุด อันเป็นผลทำให้การจำแนกหรือการทำนายข้อมูลย่อมมีความแม่นยำมากขึ้น และข้อดีอีกประการหนึ่งคือการใช้แบบจำลองตามงานวิจัยนี้จะสามารถดำเนินการได้ในแบบเวลาจริงด้วยเหตุว่าแบบจำลองใช้ระบบการทำงานแบบเปลี่ยนสถานะไปทีละขั้นในขณะที่อ่านอินพุตแต่ละตัวตามรูปแบบของออโดมาตา



รูปที่ 3.1 ลักษณะของแบบจำลองจากงานวิจัย

งานวิจัยแบ่งออกเป็น 2 ส่วนได้แก่ 1) แบบจำลองแบบใหม่และวิธีการสร้างแบบจำลองจากข้อมูลสายอักขระ และ 2) ขั้นตอนวิธีการนำแบบจำลองไปใช้สำหรับจำแนกข้อมูล ในขั้นตอนวิธีการเรียนรู้นั้นทำได้ด้วยการอ่านสายอักขระ S แล้วนำมาสร้างแบบจำลองที่ได้จากงานวิจัยนี้ เพื่อจัดเก็บสายอักขระที่แตกต่างกันและจำนวนความถี่ในการเกิดอักขระแต่ละตัว หลังจากเรียนรู้แล้วจะได้โครงสร้างข้อมูลที่เก็บสายอักขระย่อยใน S ซึ่งถูกตัดคำโดยใช้สายอักขระย่อยเอกลักษ์ที่สั้นที่สุด ซึ่งจะทำให้โครงสร้างข้อมูลมีอัตราการเติบโตของหน่วยความจำและเวลาในการประมวลผลไม่มากเกินไป เมื่อได้โครงสร้างที่มาจากการเรียนรู้แล้ว จึงนำโครงสร้างนั้นไปใช้สำหรับจำแนกข้อมูล โดยในงานวิจัยนี้จะนำเสนอขั้นตอนวิธีการคำนวณความน่าจะเป็นเพื่อจำแนกข้อมูลและเทคนิคการปรับเรียบ เพื่อให้การนำไปใช้งานจำแนกข้อมูลมีความแม่นยำ

3.2 นิยามเบื้องต้น

สายอักขระ S ใดๆสามารถแทนได้ด้วย $S = x_1x_2...x_l$โดยมีความยาวของ S เขียนแทนได้ด้วย $|S|$ เท่ากับจำนวน l สายอักขระย่อยของ S แทนด้วย $S_{i..j} = x_ix_{i+1}x_{i+2}...x_j$ เมื่อ $i, j \in I^+$ ความยาวของสายอักขระย่อยแทนด้วย $|S_{i..j}| = (j-i)+1$ เช่น $|S_{3..5}|$ เท่ากับ 3

คำเติมท้าย (suffix) ของ S แทนได้ด้วย $S_{i..l}$ เป็นคำเติมท้ายเริ่มที่ตำแหน่ง i และเซตของคำเติมท้ายแท้ (proper suffix) ของ S คือ $\{S_{i..l} | 1 < i \leq l\} \cup \{\varepsilon\}$

คำนำหน้า (prefix) ของ S แทนได้ด้วย $S_{1..j}$ เป็นคำนำหน้าที่สิ้นสุดที่ตำแหน่ง j โดยเซตของคำนำหน้าแท้ (proper prefix) ของ S คือ เซต $\{S_{1..j} | 1 \leq j < l\} \cup \{\varepsilon\}$

สายอักขระเกิดซ้ำ (repeating substring) หมายถึง สายอักขระย่ อย $S_{a..b}$ ที่พบว่ามี $S_{a..b} = S_{c..d}$ โดยที่ $1 \leq a, b, c, d \leq l$ และ $a \leq b, c \leq d$ และ $a \neq c, b \neq d$ เช่ น $S = 101001$ สายอักขระเกิดซ้ำได้แก่ 0,1,01,10

สายอักขระเอกลักษณ์ (unique substring) หมายถึง สายอักขระย่อย $S_{i..j}$ ที่เกิดขึ้นเพียงครั้งเดียว โดยที่ $|S_{i..j}| \geq 2$ เช่น $S = 101001$ สายอักขระเอกลักษณ์ได้แก่ 00,101,001,010 สายอักขระเกิดซ้ำเมื่อเพิ่มความยาวไปทางซ้ายหรือทางขวาก็ยังคงเป็นสายอักขระเกิดซ้ำ

สายอักขระเอกลักษณ์สั้นสุด (minimum unique substring) หมายถึง สายอักขระเอกลักษณ์สั้นที่สุดเท่าที่เป็นไปได้ ซึ่งมีสายอักขระย่อยที่สั้นกว่าจะเป็นสายอักขระเกิดซ้ำ เขียนแสดงได้ว่า สายอักขระเอกลักษณ์สั้นสุด $S_{i..j}$ แล้วจะมี $S_{i+1..j}$ และ $S_{i..j-1}$ เป็นสายอักขระเกิดซ้ำ ยกตัวอย่างเช่น $S = 101001$ สายอักขระเอกลักษณ์สั้นสุดคือ 00,101,010

สายอักขระเกิดซ้ำยาวสุด (maximum repeating substring) หมายถึง สายอักขระเกิดซ้ำยาวสุดเท่าที่เป็นไปได้ โดยหากเพิ่มความยาวมากขึ้นอีกเพียงอักขระตัวเดียวทางซ้ายหรือทางขวาแล้วจะกลายเป็นสายอักขระเอกลักษณ์ เขียนแสดงได้ว่า สายอักขระเกิดซ้ำยาวสุด $S_{i..j}$ โดยที่ $S_{i-1..j}$ และ $S_{i..j+1}$ เป็นสายอักขระเอกลักษณ์ เช่น $S = 101001$ สายอักขระเกิดซ้ำยาวสุด คือ 10 เพราะ 101,010 เป็นสายอักขระเอกลักษณ์

3.3 แบบจำลองที่นำเสนอ

แบบจำลองที่ใช้ในการเรียนรู้จะปรับปรุงมาจาก ออโตมาตาความน่าจะเป็นเชิงไม่กำหนด (Probaibilistic Non-deterministic finite automata) โดยเพิ่มองค์ประกอบที่สำคัญซึ่งถูกใช้ในขณะเรียนรู้และเป็นองค์ความรู้ที่แสดงข้อมูลที่สำคัญบางประการในขณะที่นำไปใช้งาน โดยองค์ประกอบที่เพิ่มเข้ามาใหม่ได้แก่ ฟังก์ชันนับความถี่ของการเปลี่ยนสถานะ ฟังก์ชันหา

คำนำหน้าของสถานะซึ่งทำหน้าที่ช่วยเก็บสายอักขระย่อยเอกลักษณ์ที่สั้นที่สุด และเซตของสถานะที่เก็บสายอักขระย่อยเอกลักษณ์ ดังนั้นแบบจำลองที่นำเสนอในงานวิจัยนี้ขอเรียกว่า *ออโตมาดาจำกัดเชิงน่าจะเป็นแบบสายอักขระย่อยเอกลักษณ์* (probabilistic unique substrings finite automata: PUSFA) ซึ่งขอเรียกโดยย่อว่า พัสฟา มีองค์ประกอบ 8 องค์ประกอบได้แก่ $M = \langle Q, \Sigma, \delta, \tau, \rho, \mu, q_0, F \rangle$ โดยมีรายละเอียดดังนี้

Q คือ เซตของสถานะ (state) โดยแต่ละสถานะจะมีฉลากชื่อกำกับไว้ ซึ่งจะใช้แทนสายอักขระข้อมูลที่ได้จากการเรียนรู้

Σ คือ เซตจำกัดของสัญลักษณ์อินพุต (input alphabet)

δ คือ ฟังก์ชันเปลี่ยนสถานะ (transition function) ซึ่งโดยเปลี่ยน สถานะ และ อินพุตไปเป็นสถานะปลายทาง แสดงได้ด้วยความสัมพันธ์ $\delta : (Q, \Sigma) \Rightarrow Q$

τ คือ ฟังก์ชันความถี่ (frequency function) จะเป็นฟังก์ชันที่เปลี่ยนจากสถานะและอินพุตไปเป็นจำนวนเต็ม แสดงได้ด้วยความสัมพันธ์ $\tau : (Q, \Sigma) \Rightarrow I^+$

μ คือ เซตของสถานะเอกลักษณ์ (a set of unique states) จะเป็นเซตที่ระบุว่าสถานะใดบ้างที่เป็นสถานะที่เก็บสายอักขระเอกลักษณ์ โดยที่ $\mu \subset Q$

ρ คือ ฟังก์ชันคำนำหน้า (prefix function) เป็นฟังก์ชันที่เปลี่ยนสถานะไปเป็นคำนำหน้าของสถานะนั้น เพื่อใช้สำหรับการเรียนรู้และจำแนกสายอักขระเอกลักษณ์ที่สั้นที่สุด
$\rho : (\mu) \Rightarrow \Sigma^*$

q_0 คือ สถานะเริ่มต้น (the initial state) มีเพียงสถานะเดียว เรียกว่า บัพราก (root node) มีฉลากติดสถานะด้วยสายอักขระว่าง ε (empty string)

F คือ เซตของสถานะสุดท้าย (a set of final states) ซึ่งเป็นสถานะที่แสดงถึงการสิ้นสุดสายอักขระที่เรียนรู้ แต่ละสถานะสุดท้ายจะมีเครื่องหมายพิเศษปิดท้าย # เป็นคำเติมท้ายของฉลากติดสถานะ และเซตของสถานะสุดท้าย $F \subseteq Q$

แบบจำลองดังกล่าวจะถูกสร้างขึ้นจากสายอักขระข้อมูล โดยใช้หลักการของสายอักขระเอกลักษณ์ที่สั้นที่สุด เพื่อที่จะตัดทอนความยาวของสายอักขระย่อยให้สั้นลง ซึ่งจะทำให้เวลาและพื้นที่ที่ใช้งานไม่เติบโตมากเกินไปเหมาะสมที่จะนำไปเรียนรู้สายอักขระขนาดไม่จำกัดได้ในการใช้งานจริง ซึ่งจะกล่าวถึงขั้นตอนวิธีในการสร้างในหัวข้อ 3. และทฤษฎีที่นำเสนอในหัวข้อจากหลักการสายอักขระที่สั้นที่สุดก่อให้เกิดแบบจำลองมีคุณสมบัติดังนี้คือ

คุณสมบัติ 3.1: สถานะแต่ละสถานะจะแบ่งออกเป็น 3 ประเภทคือ สถานะเอกลักษณ์ สถานะเอกลักษณ์สั้นสุด และ สถานะสายอักขระเกิดซ้ำ โดยมีนิยามดังนี้

นิยาม 3.1: *สถานะเอกลักษณ์* หมายถึง สถานะที่เก็บข้อมูลสายอักขระที่เกิดขึ้นครั้งเดียว หรือสายอักขระที่ขึ้นต้นด้วยอักขระนำหน้า $ และมีคำเดิมท้ายแท้ที่ยาวที่สุดเป็นสายอักขระที่เกิดซ้ำ

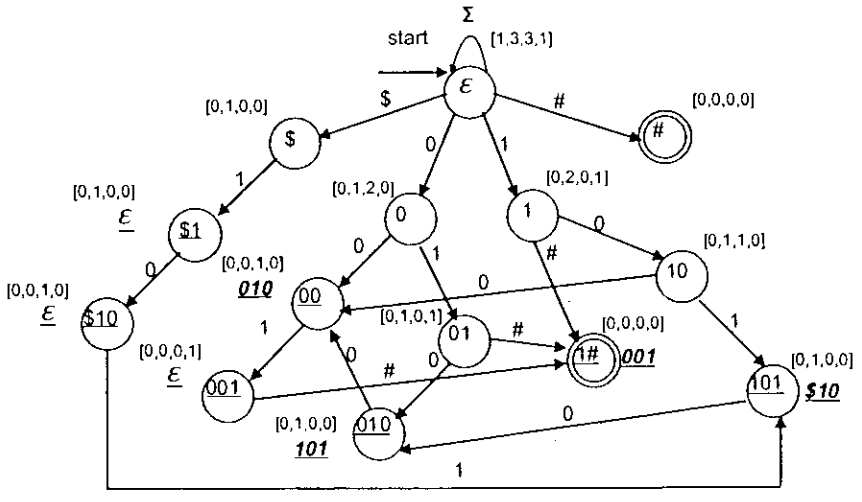
นิยาม 3.2 *สถานะเอกลักษณ์สั้นสุด* หมายถึง สถานะที่เก็บข้อมูลสายอักขระที่เกิดขึ้นครั้งเดียวที่สั้นที่สุดแทนด้วย $x_{i..j}$ โดยที่ $x_{i+1..j}, x_{i..j-1}$ เป็นสายอักขระที่เกิดซ้ำ

นิยาม 3.3 *สถานะเกิดซ้ำ* คือสถานะที่เก็บสายอักขระที่มีความยาว 1 หรือสายอักขระที่เกิดขึ้นตั้งแต่ 2 ครั้งเป็นต้นไป

คุณสมบัติ 3.2: สถานะที่มีปลายทางเดียวกันจะต้องมีคำเดิมท้ายร่วมกัน
คุณสมบัติ 3.3: สถานะที่มีป้ายชื่อสั้นกว่าจะเป็นคำเดิมท้ายของสถานะตัวที่ยาวกว่าเมื่อมีปลายทางเดียวกัน

ดูตัวอย่างแบบจำลองดังนี้

ตัวอย่าง 3.1 ออโตมาตาจำกัดเชิงน่าจะเป็นแบบสายอักขระย่อยเอกลักษณ์ที่สร้างขึ้นจาก $101001# ได้ดังรูปที่ 3.1



รูปที่ 3.2 ออโตมาตาจำกัดเชิงน่าจะเป็นแบบสายอักขระย่อยเอกลักษณ์

จากรูปที่ 3.1 แต่ละสถานะจะมีป้ายชื่อเป็นสายอักขระ w กำกับ แทนสถานะนั้นได้ด้วย $q_{\_w}$ สถานะเริ่มต้น q_0 จะถูกติดป้ายชื่อด้วย ε แทนด้วย $q_{\_\varepsilon}$ กำกับด้วยลูกศรชี้มีคำว่า *start* แต่ละสถานะมีความถี่กำกับไว้ภายในเครื่องหมาย [ ] ซึ่งระบุจำนวนครั้งของการพบอักขระ \$,0,1,# เรียงตามลำดับ เช่น ระบุความถี่ของการเกิดอักขระ [0,1,2,0] ที่สถานะป้ายชื่อ 0 แทนด้วย $q_{\_0}$ หมายความว่า มีอักขระ \$ เกิด 0 ครั้ง อักขระ 0 จำนวน 1 ครั้ง อักขระ 1 จำนวน 2 ตัว และอักขระ # จำนวน 0 ครั้ง ซึ่งแสดงว่ามี 00 จำนวน 1 ครั้งและมี 01จำนวน 2 ครั้งในสายอักขระอินพุต แต่ละสถานะที่มีป้ายชื่อขีดเส้นใต้ จะหมายถึงสถานะเอกลักษณ์ อันได้แก่ $q_{\_00}, q_{\_101}, q_{\_010}, q_{\_001}, q_{\_1\#}, q_{\_\$1}, q_{\_\$10}$ สถานะที่ป้ายชื่อไม่ขีดเส้นใต้จะเป็นสถานะเกิดซ้ำ เช่น $q_{\_1}, q_{\_0}, q_{\_10}, q_{\_01}$ สถานะเอกลักษณ์ทั่วไปได้แก่ $q_{\_\$1}, q_{\_\$10}, q_{\_001}$ จะถูกกำกับด้วยคำนำหน้าเป็นอักขระ ε แต่สถานะเอกลักษณ์สั้นสุดได้แก่ $q_{\_00}, q_{\_101}, q_{\_010}, q_{\_1\#}$ จะถูกกำกับด้วยคำนำหน้าสายอักขระที่ไม่เป็น ε แสดงอยู่ได้เครื่องหมายแสดงความถี่ $[x, x, x, x]$ แสดงด้วยสายอักขระตัวหนาขีดเส้นใต้ ซึ่งคำนำหน้าเหล่านี้จะเป็นตัวแสดงถึงสายอักขระเอกลักษณ์ที่สั้นสุดและจะถูกนำไปใช้ในการปรับเปลี่ยนโครงสร้างในการเรียนรู้ส่วนเพิ่มและจะหาได้จากฟังก์ชันคำนำหน้า ρ และสำหรับสถานะเกิดขึ้นซ้ำจะไม่มีสายอักขระกำกับ การเปลี่ยนสถานะแสดงได้ด้วยลูกศรชี้ไปยังสถานะปลายทางกำกับด้วยอักขระที่ใช้ในการเปลี่ยนสถานะ สถานะสุดท้ายจะใช้วงกลม 2 วงซ้อนกันซึ่งจะมีป้ายชื่อที่มีคำเติมท้ายเป็นอักขระพิเศษ #

3.4 การใช้งานแบบจำลองสำหรับวิเคราะห์คำสำคัญ

หลักการในการใช้งานแบบจำลองสำหรับวิเคราะห์คำสำคัญในงานวิจัยนี้จะใช้หลักการเดียวกันกับ ออโดมาตาจำกัดเชิงกำหนด โดยเริ่มต้นจากสถานะเริ่มต้นอ่านอินพุตทีละตัวแล้วเปลี่ยนสถานะไปตามฟังก์ชันเปลี่ยนสถานะ โดยไม่สนใจการเปลี่ยนสถานะด้วยอักขระใดๆใน Σ ของสถานะ $q_{\_\varepsilon}$ ซึ่งจะได้ สถานะปลายทางคำตอบเดียว หรืออาจจะใช้หลักการของ ออโดมาตาจำกัดเชิงไม่กำหนด ซึ่งจะสนใจการเปลี่ยนสถานะด้วยอักขระใดๆใน Σ ของสถานะ $q_{\_\varepsilon}$ ก็ได้ โดยจะได้สถานะปลายทางหลายคำตอบตามความเป็นไปได้ ซึ่งจะกล่าวรายละเอียดอีกครั้ง โดยผลลัพธ์สถานะแต่ละสถานะจะเป็นคำตอบโดยนัย ซึ่งมีรายละเอียดการคำนวณผลลัพธ์การเปลี่ยนสถานะดังต่อไปนี้

กำหนดให้สายอักขระ w เป็นอินพุตที่ต้องการตรวจสอบผ่านทางแบบจำลอง M โดย $w = x_1 x_2 \ldots x_n$ แล้วจะสามารถคำนวณหาผลลัพธ์ตามหลักการของออโดมาตาจำกัดเชิงกำหนดได้โดยใช้ฟังก์ชันเปลี่ยนสถานะแบบขยาย (extended transition function) ดังนี้

ให้ $w = va$ โดยที่ $v = x_1 \ldots x_{n-1}$ และ $a = x_n$ แล้ว

$$\delta^*(q_0, w) = \delta^*(q_0, va) = \delta(\delta^*(q_0, v), a)$$

ตัวอย่างเช่น ให้ $w = 1001$สามารถคำนวณได้ตามฟั งก์ชันข้างต้นหรืออาจกล่าวโดยสรุปว่าให้อ่านอินพุตแต่ละตัวแล้วเปลี่ยนสถานะไปจนกว่าจะอ่านอินพุตครบทุกตัว จะได้สถานะปลายทางแต่ละขั้นตอนเป็นผลลัพธ์ในการวิเคราะห์ ดูตัวอย่างการคำนวณดังนี้

ตัวอย่าง 3.2

กำหนดให้อินพุต $w = 1001$ และสถานะที่มีป้ายชื่อ x แทนด้วย q_x สามารถคำนวณอินพุต w ด้วยออโตมาตาตามรูปที่ 3.1 ดังนี้ เริ่มต้น ให้สถานะเริ่มต้นอยู่ที่สถานะ q_ε

1.สถานะ q_ε อ่าน 1 ได้ผลลัพธ์คือ q_1 เขียนได้โดย $\delta(q_\varepsilon,1) \rightarrow q_1$

2.จากสถานะ q_1 อ่าน 0 ได้ผลลัพธ์ สถานะ q_{10} เขียนได้โดย $\delta(q_1,0) \rightarrow q_{10}$

3.จากสถานะ q_{10} อ่าน 0 ได้ผลลัพธ์คือ สถานะ q_{00} เขียนได้โดย $\delta(q_{10},0) \rightarrow q_{00}$

4. จากสถานะ q_{00} อ่าน 1 ได้ผลลัพธ์คือ สถานะ q_{001} เขียนได้โดย $\delta(q_{00},1) \rightarrow q_{001}$

สรุปได้ว่า $\delta^*(q_\varepsilon,1001) \rightarrow q_{001}$ โดยสามารถวิเคราะห์ข้อมูลสายอักขระ 1001 ได้ดังนี้

มี 1 เช่นเดียวกับข้อมูลสายอักขระซึ่งเกิดซ้ำจำนวน 3 ครั้ง (ดูจากฟังก์ชันความถี่ที่อยู่ที่สถานะ q_1)

มี 10 เช่นเดียวกับข้อมูลสายอักขระซึ่งเกิดซ้ำจำนวน 2 ครั้ง

มี 00 เช่นเดียวกับข้อมูลสายอักขระ ซึ่งเป็นสายอักขระเอกลักษณ์ที่สั้นที่สุด (เพราะ q_{00} เป็นสถานะเอกลักษณ์ที่สั้นที่สุด)

มี 001 เช่นเดียวกับข้อมูลสายอักขระ ซึ่งเป็นสายอักขระเอกลักษณ์

สายอักขระ 1001 เป็นสายอักขระย่อยของข้อมูลที่เรียนรู้ เพราะมีสถานะรองรับการเปลี่ยนสถานะเมื่ออ่าน 1001 ครบทุกตัว □

สำหรับการตรวจสอบคำเติมท้าย สามารถทำได้โดย ต่ออักขระพิเศษ # ที่สายอักขระ w เช่น 1001# ซึ่งหากเปลี่ยนสถานะไปจนถึงสถานะสุดท้ายได้ แสดงว่าสายอักขระ w เป็นคำเติมท้ายของชุดข้อมูล

ในกรณีที่ใช้หลักการคำนวณแบบออโตมาตาจำกัดเชิงไม่กำหนด (non-deterministic finite autmata) สำหรับประยุกต์ใช้ในการวิเคราะห์คำสำคัญที่มีหลายคำรวมๆกันอยู่ในสายอักขระ ซึ่งหากใช้การคำนวณในแบบของออโตมาตาจำกัดเชิงกำหนดแล้วในบางครั้งอาจจะไม่มีเส้นทางไปต่อ ยกตัวอย่างเช่น สายอักขระ 0011 จะมีส่วน 00, 01 และ 001 ที่ยอมรับได้ใน ออโตมาดาดามรูปที่ 3.1 แต่ไม่มี 11 ทำให้สถานะของออโตมาดาไม่มีเส้นทางไปต่อและอ่านอินพุตได้ไม่ครบ แต่หากใช้หลักการของเอ็นเอฟเอ แล้วจะทำให้มีเส้นทางไปต่อได้จนจบอินพุต ซึ่งได้

ปลายทางที่สถานะ 1 ทำให้สามารถวิเคราะห์สายอักขระ 0011 ได้ว่ามี 00, 01, 001, 1 ที่เกิดขึ้น โดยมีหลักการคำนวณดังต่อไปนี้

ให้การเปลี่ยนสถานะของ $q_{\_\varepsilon}$ สำหรับทุกตัวอักษรวนเข้าสู่ตัวเองด้วยอีกเส้นทางหนึ่ง ซึ่งจะทำให้ การเปลี่ยนสถานะจากสถานะเริ่มต้นไปได้ 2 เส้นทางคือ เส้นทางปกติ และเส้นทางกลับเข้าสู่ตัวจุดเริ่มต้นใหม่อีกครั้ง ทำให้เกิดผลลัพธ์สถานะปลายทางได้หลายสถานะ ซึ่งแทนได้ด้วยเซต โดยเขียนแสดงได้ดังนี้

ให้ $w = va$ โดยที่ $v = x_1x_2\ldots x_{n-1}$และ $a = x_n$ และ ให้ $\delta^*(q_0, v) = \{p_1, p_2, \ldots, p_k\}$แล้ว

$$\delta^*(q_0, va) = \bigcup_{i=1}^{k} \delta(p_i, a)$$

ตัวอย่าง 3.3

กำหนดให้อินพุต $w = 0011$ สามารถคำนวณอินพุต w ด้วยออโตมาดาตามรูปที่ 3.1 ตามหลักการของเอ็นเอฟเอดังนี้ เริ่มต้น ให้สถานะเริ่มต้นอยู่ที่สถานะ $q_{\_\varepsilon}$

1.สถานะ $q_{\_\varepsilon}$ อ่าน 0 ได้ผลลัพธ์คือ $q_{\_\varepsilon}, q_{\_0}$ แสดงได้ด้วย $\delta(q_{\_\varepsilon},0) \rightarrow \{q_{\_\varepsilon}, q_{\_0}\}$

2.สถานะ $\{q_{_\varepsilon}, q_{_0}\}$อ่าน 0 ได้ผลลัพธ์คือ $q_{\_\varepsilon}, q_{\_0}, q_{\_00}$ แสดงได้ด้วย $\delta(\{q_{\_\varepsilon}, q_{\_0}\},0) \rightarrow \{q_{\_\varepsilon}, q_{\_0}, q_{\_00}\}$

3.สถานะ $\{q_{\_\varepsilon}, q_{\_0}, q_{\_00}\}$ อ่าน 1 ได้ผลลัพธ์คือ สถานะ $q_{\_\varepsilon}, q_{\_1}, q_{\_01}, q_{\_001}$ แสดงได้ด้วย $\delta(\{q_{\_\varepsilon}, q_{\_0}, q_{\_00}\},1) \rightarrow \{q_{\_\varepsilon}, q_{\_1}, q_{\_01}, q_{\_001}\}$

4.สถานะ $\{q_{\_\varepsilon}, q_{\_1}, q_{\_01}, q_{\_001}\}$ อ่าน 1 ได้ผลลัพธ์คือ สถานะ $q_{\_\varepsilon}, q_{\_1}$ แสดงได้ด้วย $\delta(\{q_{\_\varepsilon}, q_{\_1}, q_{\_01}, q_{\_001}\},1) \rightarrow \{q_{\_\varepsilon}, q_{\_1}\}$

เมื่อพิจารณาจากสถานะในแต่ละขั้นตอนที่ผ่านมา หากพิจารณาสถานะปลายทางที่ยาวที่สุดซึ่งมีเส้นทางไปได้แก่ $q_{_0}, q_{_00}, q_{_001}, q_{_1}$ตามลำดับ ซึ่งสามารถวิเคราะห์สายอักขระ 0011 ในเบื้องต้นได้ดังนี้

มี 0 เช่นเดียวกับข้อมูลสายอักขระซึ่งเคยเกิดซ้ำจำนวน 3 ครั้ง (ดูจากฟังก์ชันความถี่ที่อยู่ที่สถานะ $q_{\_0}$)

มี 00 เช่นเดียวกับข้อมูลสายอักขระ ซึ่งเป็นสายอักขระเอกลักษณ์ที่สั้นที่สุด

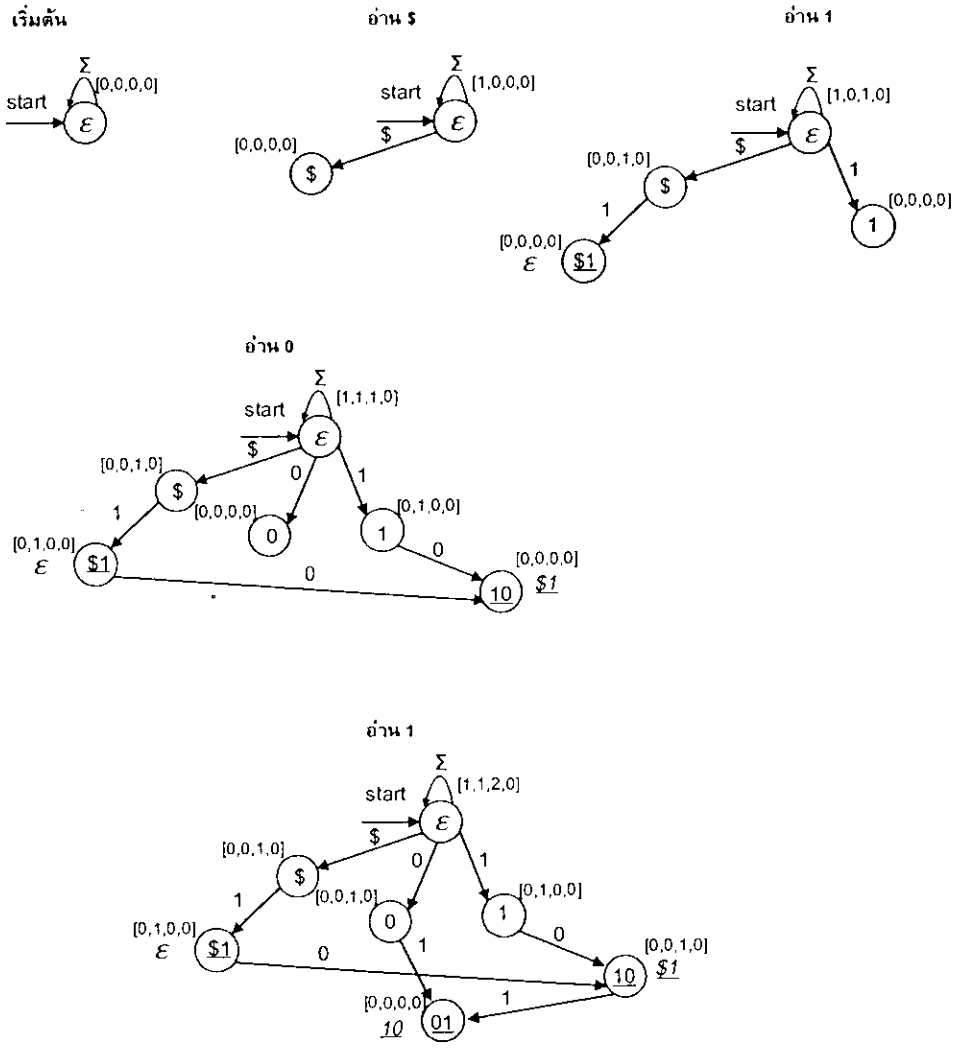
มี 001 เช่นเดียวกับข้อมูลสายอักขระ ซึ่งเป็นสายอักขระเอกลักษณ์

มี 1 เช่นเดียวกับข้อมูลสายอักขระซึ่งเกิดซ้ำจำนวน 3 ครั้ง

สรุปได้ว่าสายอักขระ 0011 ไม่เป็นสายอักขระย่อยของข้อมูลที่เรียนรู้ แต่พบสายอักขระย่อยที่เหมือนกันที่ยาวที่สุดคือ 001 □

3.5 ขั้นตอนวิธีในการสร้างแบบจำลอง

หลักการในการเรียนรู้คือ ให้เพิ่มอักขระพิเศษ $ นำหน้าสายอักขระที่ต้องการจะเรียนรู้ และเพิ่มอักขระพิเศษ # ตามหลังสายอักขระที่ต้องการเรียนรู้เช่น $w = 101001$ จะได้สายอักขระสำหรับเรียนรู้คือ $w = \$101001\#$ เพื่อแยกความแตกต่างอักขระที่ขึ้นต้นและลงท้าย จากนั้นเริ่มต้นให้สร้างสถานะเริ่มต้นและมีการเปลี่ยนสถานะทุกตัวอักษรวนเข้าสู่ตัวเอง แล้วทำการอ่านอักขระแต่ละตัวในสายอักขระอินพุต หากอักขระที่อ่านมาไม่มีเส้นทางไปนอกจากวนเข้าสู่สถานะเริ่มต้นแล้ว ให้ทำการสร้างสถานะใหม่ขึ้นเพื่อรองรับอักขระตัวนั้น โดยให้มีป้ายชื่อเท่ากับสถานะเดิมต่อกันกับอักขระอินพุตที่อ่านมา แต่หากมีเส้นทางไปให้นับความถี่เพิ่มขึ้นอีกหนึ่งในเส้นทางนั้น อย่างไรก็ตามสถานะปลายทางในแต่ละครั้งที่อ่านอักขระ 1 ตัวจะถูกนำไปเป็นสถานะที่พิจารณาในรอบถัดไปเรียงตามลำดับความยาวน้อยไปมาก เรียกว่า สถานะรอทำการ (active state) สถานะรอทำการนี้จะมีคุณสมบัติคือ ตัวที่สั้นกว่าจะเป็นคำเติมท้ายของตัวที่ยาวกว่าและมีจำนวนเท่ากับ ความยาวของตัวที่ยาวที่สุด + 1 เช่น สมมุติให้สถานะรอทำการตัวที่มีความยาวที่สุดคือ $q_{\_101}$ จะมีสถานะรอทำการที่เหลือคือ $q_{\_01}, q_{\_1}, q_{\_\varepsilon}$ ซึ่งมีจำนวนสถานะรอทำการทั้งหมด 4 ตัว โดยรอทำการเรียงลำดับจากความยาวน้อยไปมาก สถานะที่ถูกสร้างขึ้นใหม่ในแต่ละรอบที่มีความยาวตั้งแต่ 2 อักขระเป็นต้นไปจะถือว่าเป็นสถานะเอกลักษณ์ (unique state) สถานะเอกลักษณ์จะถูกสร้างขึ้นเพื่อรองรับการเปลี่ยนสถานะจากสถานะรอทำการตัวใดตัวหนึ่ง และสามารถเป็นปลายทางให้กับสถานะรอทำการตัวอื่นที่มีความยาวเท่ากันหรือมากกว่าสถานะเอกลักษณ์ด้วย โดยสถานะรอทำการที่มีปลายทางไปยังสถานะเอกลักษณ์จะต้องเพิ่มคำนำหน้าบันทึกไว้ที่สถานะเอกลักษณ์นั้นด้วยชื่อสถานะรอทำการที่ยาวที่สุดในรอบดำเนินการเดียวกันนั้น เพื่อใช้สำหรับตรวจสอบและปรับสถานะเอกลักษณ์ไปเป็นสถานะสายอักขระเกิดซ้ำในกรณีที่พบอินพุตใหม่เข้ามาทำให้ สถานะเอกลักษณ์เปลี่ยนไปเป็นสถานะสายอักขระเกิดซ้ำ ดูรูปที่ 3.3 เป็นการแสดงการสร้างแบบจำลองด้วยสายอักขระ $101 สถานะรอทำการจะแสดงด้วยวงกลมทึบ



รูปที่ 3.3 ออโตมาตาจำกัดเชิงน่าจะเป็นแบบสายอักขระย่อยเอกลักษณ์สร้างจากสายอักขระ $101

ในกรณีที่สถานะรอทำการตัวใดตัวหนึ่งสมมติมีป้ายชื่อ $x_{j..l}$ เมื่อ $x_{j..l}$ แทนสายอักขระย่อยของอินพุตตำแหน่งที่ j ถึง l โดยแทนสถานะนั้นด้วย $q_{\_x_{j..l}}$ มีเส้นทางไปยังสถานะเอกลักษณ์ $q_{\_x_{k..l+1}}$ ด้วยอักขระอิ นพุต c ใดๆอยู่ก่อนแล้ว ให้ทำการเพิ่มความถี่ของสถานะ $q_{\_x_{j..l}}$ ของเส้นทาง c และกำหนดให้สถานะเอกลักษณ์ $q_{\_x_{k..l+1}}$ นั้นจะต้องเป็นเป้าหมายปลายทางให้กับสถานะรอทำการที่ยาวกว่า $q_{\_x_{j..l}}$ อันได้แก่ คือ $q_{\_x_{j-n..l}}$ เมื่อ $n \geq 1$ ซึ่งหากพบว่าคำนำหน้าอันได้แก่ $x_{i..l}$ ของสถานะเอกลักษณ์ $q_{\_x_{k..l+1}}$ นั้นไม่เป็นคำเติมท้ายของสถานะรอทำการ $q_{_x_{j-n..l}}$ที่กำลั งจะสร้างเส้นทางไปถึง แสดงว่าสถานะเอกลักษณ์ $q_{\_x_{k..l+1}}$ นั้นจะต้องเปลี่ยนไปเป็นสถานะสายอักขระเกิดซ้ำ และจะต้องทำการแบ่ง (split) สถานะเอกลักษณ์ $q_{_x_{k..l+1}}$นั้น ด้ วยการปรับเปลี่ยนเส้นทางสถานะก่อนหน้า (previous state) ของ $q_{\_x_{k..l+1}}$ ซึ่งสามารถหาได้จากคำนำหน้า $x_{i..l}$ จะได้ตั้งแต่สถานะ $q_{\_x_{k-1..l}}, q_{\_x_{k-2..l}}, q_{\_x_{k-3..l}}, \cdots q_{\_x_{i..l}}$ ซึ่งแต่ละสถานะดังกล่าวเดิมจะมีเส้นทางมาถึงสถานะ $q_{\_x_{k..l+1}}$ ให้ลบเส้นทางเดิม แล้วเปลี่ยนเส้นทางใหม่โดยพิจารณาเงื่อนไขที่จะต้องดำเนินการดังต่อไปนี้

1. สร้างสถานะปลายทางใหม่เพื่อรองรับเส้นทางให้กับสถานะรอทำการ $q_{\_x_{j-1..l}}$ โดยให้สถานะใหม่มีป้ายชื่อเป็น $q_{\_(x_{j-1..l})c}$ พร้อมกับเพิ่มข้อมูลความถี่ให้$q_{\_x_{j-1..l}}$ โดยกำหนดให้สถานะใหม่ $q_{\_(x_{j-1..l})c}$ เป็นสถานะเอกลักษณ์และเป็นสถานะเป้าหมายสำหรับสถานะรอทำการ $q_{\_x_{j-n..l}}$ ตัวที่ยาวกว่าต่อไปในรอบนั้น และเพิ่มเป็นสถานะรอทำการให้กับรอบถัดไป

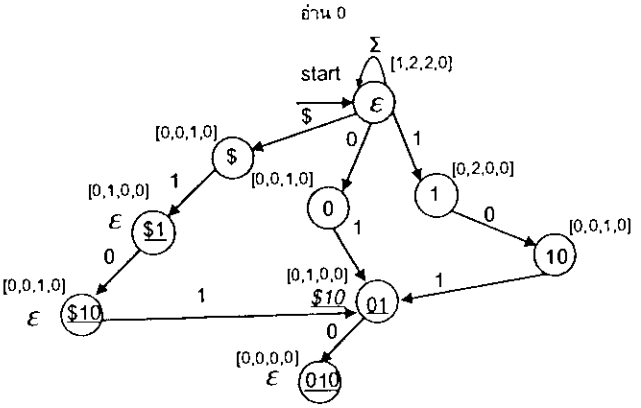
2. สถานะก่อนหน้าได้แก่ $q_{\_x_{k-1..l}}, q_{\_x_{k-2..l}}, q_{\_x_{k-3..l}}, \cdots q_{\_x_{k-m..l}}$ เฉพาะป้ายชื่อที่มีความยาวน้อยกว่าหรือเท่ากันกับ $x_{j-1..l}$ จากข้อ 1 โดยมี $|x_{k-m..l}| = |x_{j-1..l}|$ ให้ทำการสร้างสถานะใหม่มารองรับ โดยให้สถานะใหม่มีป้ายชื่อเป็นชื่อเดิมต่อกันกับ c ได้แก่ $q_{\_x_{k-1..l}c}, q_{\_x_{k-2..l}c}, q_{\_x_{k-3..l}c}, \cdots q_{\_x_{k-m..l}c}$ เช่น สถานะเดิมเป็น $q_{\_10}$ และมี $c = 1$ ให้สร้างสถานะใหม่เป็น $q_{\_101}$ และกำหนดให้ $q_{\_(x_{k-m..l})c}$ เป็นสถานะเป้าหมายสำหรับสถานะก่ อนหน้าตัวที่ยาวกว่า $x_{j-1..l}$ โดยให้สถานะใหม่ทุกตัวที่เป็นคำเติมท้ายของ $q_{\_(x_{j-1..l})c}$ ที่สร้างจากข้อ 1 เป็นสถานะสายอักขระเกิดซ้ำ เพราะเนื่องจากสถานะ $q_{\_(x_{j-1..l})c}$ เป็นสถานะเอกลักษณ์สั้นสุดเพราะเพิ่งสร้างขึ้นใหม่ ดังนั้นคำเติมท้ายแท้ทุกตัวของสถานะนั้นจะเป็นสถานะเกิดซ้ำ และสถานะใหม่ทุกสถานะที่เป็นคำเติมท้ายของ $q_{\_(x_{j-1..l})c}$ จะถูกจัดเก็บเป็นสถานะรอทำการสำหรับการอ่านอินพุตรอถัดไป สถานะใหม่ที่สร้างขึ้นจะต้องคัดลอกการเปลี่ยนสถานะ ความถี่และปลายทางจากสถานะเอกลักษณ์ $q_{_x_{k..l+1}}$ที่ถูกเปลี่ยนมาเป็นของตัวเองด้วย และให้สถานะปลายทางของสถานะ $q_{\_x_{k..l+1}}$ จะต้องเพิ่มสถานะใหม่ตัวที่ยาวที่สุดเป็นคำนำหน้า (สถานะปลายทางจะมีสถานะเดียวเพราะ $q_{\_x_{k..l+1}}$ เป็นเอกลักษณ์มาก่อน)

3. สถานะใหม่ $q_{\_(x_{k-m..l})c}$ จากข้อ 2 จะเป็นสถานะเอกลักษณ์และเป็นสถานะเป้าหมายปลายทางใหม่ให้กลุ่มของสถานะก่อนหน้าที่ถูกแบ่งเส้นทางออกมาตัวที่มีความยาวมากกว่า $x_{j-1..l}$ ดังนั้น สถานะก่อนหน้าที่มีความยาวมากกว่า $x_{j-1..l}$ จะทำการสร้างเส้นทางไปยังปลายทางคือ $q_{\_(x_{k-m..l})c}$ รวมถึงเก็บคำนำหน้าของสถานะ $q_{\_(x_{k-m..l})c}$ ด้วยสถานะก่อนหน้าที่ยาวที่สุดที่ซื้อไป

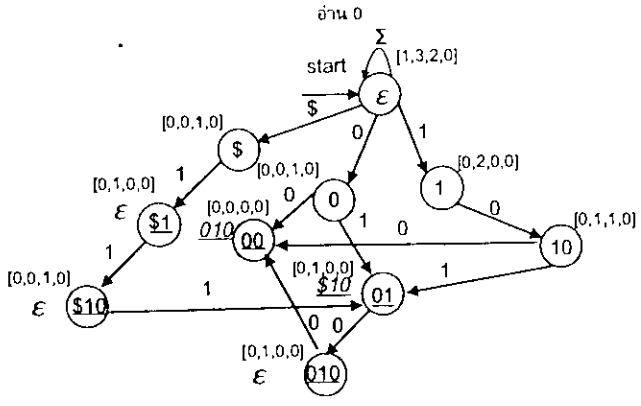
ดูการขั้นตอนวิธีจาก รูปที่ 3.4-3.7 ซึ่งจะแสดงการสร้างแบบจำลองจากสายอักขระ \$101001# โดยยกเอาข้อมูลที่สร้างไปแล้วด้วย \$101 จากรูปที่ 3.3 มาสร้างต่อ ด้วยการอ่าน 001# เพิ่มเข้ามา เริ่มต้นจากการอ่าน 0 แสดงในรูปที่ 3.4 โดยมีสถานะรอทำการเดิมคือ $q_{\_\varepsilon}, q_{\_1}, q_{\_01}$ เมื่อ $q_{\_\varepsilon}, q_{\_1}$ อ่าน 0 จะได้สถานะปลายทางคือ $q_{\_\varepsilon}, q_{\_0}, q_{\_10}$ ตามลำดับ ซึ่งสถานะ $q_{\_10}$ จะเป็นเป้าหมายให้กับสถานะ $q_{\_01}$ แต่เนื่องจากพบว่าคำนำหน้าของสถานะ $q_{\_10}$ ได้แก่ \$1 นั้นไม่เป็นคำเติมท้ายของ $q_{\_01}$ ทำให้ต้องเปลี่ยนสถานะ $q_{\_10}$ เป็นสถานเกิดซ้ำและแบ่งเส้นทางออกตามขั้นตอนวิธีข้อ 2 โดยสร้าง $q_{\_010}$ เป็นสถานะเอกลักษณ์ตัวใหม่ และสถานะก่อนหน้าของ $q_{\_10}$ คือ $q_{\_\$1}$(สถานะก่อนหน้าจะมีความยาวไม่ต่ำกว่าสถานะที่สนใจ) และสร้างสถานะใหม่ให้กับ $q_{\_\$1}$ ได้เป็น $q_{\_\$10}$ แล้วคัดลอกความถี่และสถานะปลายทางเดิม

เมื่อได้ผลลัพธ์จากการอ่าน 0 ตามรูปที่ 3.4 แล้วทำการอ่าน 0 ต่ออีกตัวหนึ่งดูรูปที่ 3.5 สถานะรอทำการ $q_{\_0}$ เมื่ออ่าน 0 จะทำการสร้างสถานะใหม่ $q_{\_00}$ ขึ้นมารองรับ และ $q_{\_00}$ จะเป็นสถานะเป้าหมายสำหรับให้สถานะรอทำการที่ยาวกว่าคือ $q_{\_10}$ และ $q_{\_010}$ สร้างเส้นทางไปยังสถานะ $q_{\_00}$ ทำให้เกิดสถานะรอทำการสำหรับรอบถัดไปคือ $q_{\_\varepsilon}, q_{\_0}, q_{\_00}$

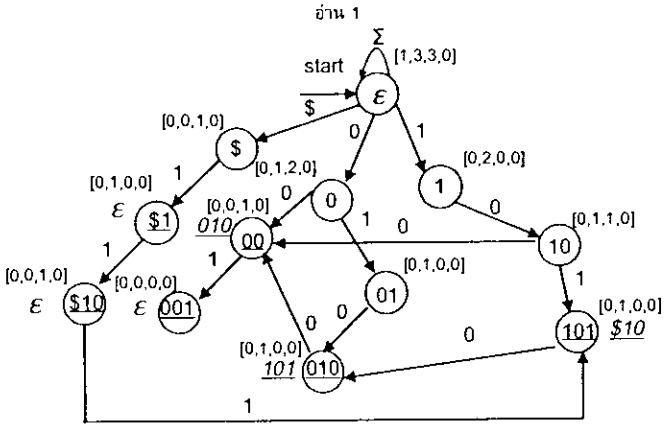
จากนั้นจะอ่านอักขระ 1 ดูรูปที่ 3.6 โดยพิจารณาสถานะรอทำการที่ได้จากรอบก่อนได้แก่ $q_{\_\varepsilon}, q_{\_0}, q_{\_00}$ ซึ่งจะได้สถานะปลายทางสำหรับ $q_{\_\varepsilon}, q_{\_0}$ คือ $q_{\_1}, q_{\_01}$ ตามลำดับเพราะมีเส้นทางอยู่ก่อนหน้าแล้ว โดยจะทำการเพิ่มความถี่ในกรณีที่มีเส้นทางอยู่ ซึ่งเมื่อดำเนินการจนถึงสถานะ $q_{\_01}$ พบว่า $q_{\_01}$ นั้นเป็นสถานะเอกลักษณ์ จึงอาจจะเป็นเป้าหมายให้กับ $q_{\_00}$ ได้ แต่เนื่องจากคำนำหน้าของ $q_{\_01}$ นั้นคือ \$10 ซึ่งไม่เป็นคำเติมท้ายของ 00 ทำให้สถานะ $q_{\_01}$ จะต้องถูกเปลี่ยนเป็นสถานะเกิดซ้ำและต้องแบ่งออก โดยทำการสร้างสถานะใหม่ต่อจาก $q_{\_00}$ เป็น $q_{\_001}$ ทำการเก็บ $q_{\_001}$ เป็นสถานะรอทำการและจากคำนำหน้า \$10 พบว่าสถานะก่อนหน้า 2 สถานะได้แก่ 10 และ \$10 โดยทำการสร้างสถานะใหม่เฉพาะ $q_{\_10}$ เนื่องจากเป็นสถานะก่อนหน้าที่มีความยาวเท่ากันกับ $q_{\_01}$ ได้สถานะใหม่เป็น $q_{\_101}$ และให้ $q_{\_101}$ เป็นเป้าหมายให้ $q_{\_\$10}$ ต่อไป จะได้สถานะรอทำการทั้งหมดได้แก่ $q_{\_\varepsilon}, q_{\_1}, q_{\_01}, q_{\_001}$



รูปที่ 3.4 ออโตมาตาจำกัดเชิงน่าจะเป็นแบบสายอักขระย่อยเอกลักษณ์สร้างจากสายอักขระ $1010

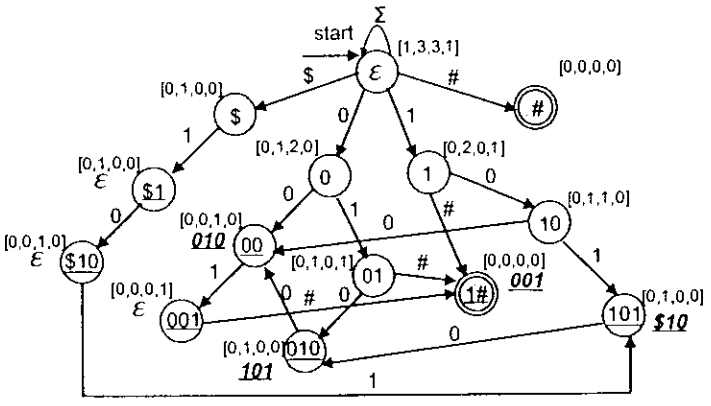


รูปที่ 3.5 ออโดมาตาจำกัดเชิงน่าจะเป็นแบบสายอักขระย่อยเอกลักษณ์สร้างจากสายอักขระ $10100



รูปที่ 3.6 ออโตมาตาจำกัดเชิงน่าจะเป็นแบบสายอักขระย่อยเอกลักษณ์สร้างจากสายอักขระ $101001

สำหรับอักขระพิเศษตัวสุดท้ายได้แก่ # จะพิจารณาจากสถานะรอทำการตามรูป 3.6 ได้แก่ $q_{\_\varepsilon}, q_{\_1}, q_{\_01}, q_{\_001}$ โดย $q_{\_\varepsilon}, q_{\_1}$ จะสร้างสถานะใหม่ $q_{\_\#}$ และ $q_{\_1\#}$ ซึ่ง $q_{\_1\#}$ จะเป็นสถานะเป้าหมายให้กับสถานะรอทำการที่เหลือคือ $q_{\_01}, q_{\_001}$ แล้วเก็บคำนำหน้าที่ $q_{\_1\#}$ ด้วยตัวที่ยาวที่สุดคือ 001 แล้วให้ $q_{\_\#}$ และ $q_{\_1\#}$ เป็นสถานะยอมรับ จะได้ผลลัพธ์ตามรูป 3.7



รูปที่ 3.7 ออโตมาตาจำกัดเชิงน่าจะเป็นแบบสายอักขระย่อยเอกลักษณ์สร้างจากสายอักขระ $101001#

จากขั้นตอนวิธีที่ได้อธิบายตามที่กล่าวมาแล้วนั้นสามารถนำมาเขียนโปรแกรมได้โดยแสดงเป็นรหัสเทียมตามรูป 3.8 และ 3.9 ซึ่งแบ่งออกเป็นสองส่วนคือ ส่วนของฟังก์ชันหลักในการสร้างแบบจำลองแสดงในรูป 3.8 และฟังก์ชันแบ่งสถานะ (splitting) แสดงในรูปที่ 3.9 ซึ่งสามารถวิเคราะห์ประสิทธิภาพเชิงเวลาได้ในส่วนของการวนซ้ำฟังก์ชันหลักจำนวน $k \cdot n$ เมื่อ k คือความยาวของสายอักขระเอกลักษณ์เฉลี่ยที่เกิดขึ้น และ n คือความยาวของสายอักขระอินพุต และในส่วนการวนซ้ำตามจำนวนสถานะก่อนหน้า ซึ่งหากเฉลี่ยว่าสายอักขะเอกลักษณ์ที่เกิดขึ้นมีความยาวเท่ากันทั้งหมดแล้ว $p \approx 1$ ซึ่งประมาณได้เท่ากับ $k \cdot p \cdot n \approx k \cdot n$ และประสิทธิภาพเชิงพื้นที่จะใช้จำนวนสถานะของออโตมาตาที่สร้างขึ้นเทียบกับความยาวของสายอักขระอินพุต โดยให้ k คือความยาวของสายอักขระเอกลักษณ์เฉลี่ยที่เกิดขึ้นและ m คือจำนวนของสายอักขระเอกลักษณ์ที่แตกต่างทั้งหมดแล้วจะได้ว่า จำนวนสถานะเอกลักษณ์ 1 สถานะรวมกับจำนวนสถานะสายอักระเกิดซ้ำจะได้ $k + (k-1) + (k-2) \ldots + 2 + 1$ หรือซึ่งเท่ากับผลรวมเลขคณิตของ i เมื่อ $i = 1$ ถึง k ได้เท่ากับ $(k+1) \cdot k/2 \approx k^2/2$ ดังนั้นจำนวนสถานะที่เกิดในกรณีที่แย่สุดซึ่งไม่มีสถานะเกิดซ้ำที่เหมือนกันเลยจะเท่ากับ $k^2/2$ คูณด้วยจำนวนสายอักขระเอกลักษณ์ที่แตกต่างทั้งหมด จะได้เท่ากับ $m \cdot k^2/2$ แต่ในความเป็นจริงแล้วสายอักขระที่เกิดขึ้นจะมีคำเติมท้ายที่ซ้ำกัน ดังนั้นจำนวนสถานะส่วนที่ซ้ำกันจะไม่ต้องถูกสร้างขึ้นใหม่ ทำให้ในทางปฏิบัติแล้วจำนวนสถานะที่เพิ่มขึ้นนั้นไม่มากเท่ากับ $m \cdot k^2/2$ ซึ่งหากพิจารณาว่า สายอักขระอินพุตมีความยาว n และอักขระทุกตัวมีความแตกต่างเช่น $abcdefghij \ldots$ แล้วจะได้ว่าค่า m จะเท่ากับ n โดยมีค่า $k = 2$ ซึ่งจากประสิทธิภาพเชิงเวลาเท่ากับ $k \cdot n$ และพื้นที่เท่ากับ $(k \cdot n)/2$ เมื่อแทนค่า $k = 2$ แล้วจะได้ว่าประสิทธิภาพเชิงเวลาดีที่สุดทำได้ $2 \cdot n$ และเชิงพื้นที่ดีที่สุด $2 \cdot n$ แต่ในความจริงหากชุดตัวอักษรมีขนาดจำกัดแล้วจะเกิดการซ้ำทำให้ไม่ถึง $2 \cdot n$ ในการใช้งานจริงนั้นค่า k จะขึ้นอยู่กับข้อมูลที่แตกต่างกันไป อย่างไรก็ดีหากวิเคราะห์ค่า k ในกรณีที่อักขระเหมือนกันทุกตัวเช่น $111 \cdots 11$ $k = 1,2,3,\ldots,n$ จะทำงานในแต่ละรอบทั้งหมด $1 + 2 + 3 + \ldots + n$ โดยไม่ต้องพิจารณาค่า m และ n จะได้ว่าเวลาที่ใช้จะประมาณ $n \cdot (n-1)/2$ ซึ่งจะอยู่ในรูปฟังก์ชัน n^2 และพื้นที่ที่ใช้จะสร้างสถานะใหม่เท่ากับจำนวน n ในกรณีสายอักขระเอกลักษณ์มี $k = n/2$ ซึ่งเป็นค่าเฉลี่ยเท่ากันทุกสายแล้วสถานะก่อนหน้าของแต่ละตัวจะมีความยาวไม่ต่างกันกับสถานะเอกลักษณ์ดังนั้น $p \approx 1$ จะทำให้ ประสิทธิภาพเชิงเวลาอยู่ในรูป $k \cdot n \approx n^2/2$ และประสิทธิภาพเชิงพื้นที่สามารถประมาณได้ว่า เมื่อ $p \approx 1$แล้วจะทำให้การแบ่งสถานะใหม่แต่ละรอบจะมีเพียง 1 สถานะ และในแต่ละครั้งที่อ่านอักขระอินพุตจะสร้างสถานะใหม่เพียง 1 ตัวซึ่งจะทำให้การสร้างสถานะใหม่เมื่ออ่านอินพุต n ตัวทั้งหมดจะประมาณได้ n สถานะ ประสิทธิภาพเชิงเวลาและเชิงพื้นที่ทำได้ดีที่สุด $O(n)$ ทำได้แย่ที่สุดไม่เกิน $O(n^2)$

Algorithm 1 Building PUSFA

Input: $inputStrings$
Output: Probabilistic Unique Substring Finite Automata M
for x_i **in** $inputStrings$
{ **for** $astate$ **in** $activeStates$
{ **if** $\tau_M(astate, x_i)$== 0
{ **if** $target == null$
{ $newstate = astate + ch$
$\tau_M(astate, x_i) = 1, \delta_M(astate, x_i) = newstate$
if $|newstate| \geq 2$
{ **add** $newstate$ **to** μ_M
$target = newstate$ }
add $newstate$ **to** $nextActiveStates$ }}
else
{ **if** $\rho_M(target)$ **is a suffix of** $astate$
{ $\tau_M(astate, x_i) = 1, \delta_M(astate, x_i) = target$
if $|astate| > \rho_M(target)$
$\rho_M(target) = astate$ }
else
{ $newstate = astate + ch$
add $newstate$ **to** μ_M
$\tau_M(astate, x_i) = 1, \delta_M(astate, x_i) = newstate$
$splitNode(target, astate, ch)$
$target = newstate$}
add $target$ **to** $nextActiveStates$ }}
else
{ $nextstate = \delta_M(astate, x_i)$
$\tau_M(nextstate, x_i) += 1$
if $nextstate \in \mu_M$ **and** $\rho_M(nextstate) \neq null$
$target = nextstate$
if $\tau_M(nextstate, x_i) > 1$
$\mu_M - nextstate$
add $nextstate$ **to** $nextActiveStates$ }}
copy $nextActiveStates$ **to** $activeStates$ }

รูปที่ 3.8 รหัสเทียมสำหรับสร้างออโตมาตาจำกัดเชิงน่าจะเป็นแบบสายอักขระย่อยเอกลักษณ์

Algorithm 2 SplitNode

```
function splitNode(splitstate, activestate1,inputchar)
{       target = null
        μ_M − splitstate
        add splitstate to nextActiveStates
        for statelabel in suffixOf(p_M(splitstate))
        {       if |statelabel| ≥ |splitstate|
                {       if target = = null
                        {       newnode = statelabel + inputchar
                                δ_M(statelabel, x_i) = newnode
                                add newnode to μ_M
                                if statelabel is a suffix of activestate1
                                {       add newnode to nextActiveStates
                                        μ_M − newnode
                                }
                                if |statelabel| = = |activestate1|
                                {       target = newnode
                                }
                                for ch in Σ_M
                                {       if τ_M(splitstate, ch) > 0
                                        {       nextofsplitstate = δ_M(splitstate, ch)
                                                τ_M(newnode, ch) = τ_M(splitstate, ch)
                                                if |newnode| > |ρ_M(nextofsplitState)|
                                                {       ρ_M(nextofsplitState) = newnode }
                                                δ_M(newnode, ch) = nextofsplitstate }}}
                        else
                        {       δ_M(statelabel, inputchar) = target
                                add target to μ_M
                                if |statelabel| > |p_M(target)|
                                {       ρ_M(target) = statelabel }}}}
```

รูปที่ 3.9 รหัสเทียมฟังก์ชันแบ่งสถานะ

ทฤษฎีบท 1 หากสายอักขระเอกลักษณ์สั้นที่สุดและสายอักขระเกิดซ้ำมีอยู่จริงในชุดข้อมูล จะมีสถานะสายอักขระเอกลักษณ์สั้นที่สุดและสายอักขระเกิดซ้ำของแบบจำลองที่สร้างขึ้นจากขั้นตอนวิธีตามรูปที่ 3.8 ซึ่งมีป้ายชื่อเป็นสายอักขระเอกลักษณ์สั้นที่สุดและสายอักขระเกิดซ้ำรองรับเสมอ

พิสูจน์ ให้สายอักขระ $x_{1..n}$ เป็นสายอักขระข้อมูลที่ใช้สร้างแบบจำลองแล้ว ย่อมมีสายอักขระ $x_{i..j}$ ซึ่งเกิดขึ้นครั้งเดียวสั้นที่สุดเท่าที่เป็นไปได้คือ ความยาวของ $x_{i..j}$ เท่ากับ 2 และหากว่า $x_{i+1..j+1}$ เกิดขึ้นครั้งเดียวและความยาวสั้นที่สุดเท่าที่เป็นไปได้คือ 2 แล้ว $x_{i..j}$ จะเป็นคำนำหน้าเพียงตัวเดียวของ $x_{i+1..j+1}$ แต่หากพบว่ามี $x_{m+1..n+1} = x_{i+1..j+1}$ ซึ่งทำให้ $x_{i..j}$ ไม่เป็นคำนำหน้าตัวเดียวของ $x_{i+1..j+1}$ ดังนั้น $x_{i+1..j+1}$ ย่อมเป็นสายอักขระที่เกิดซ้ำ และแน่นอนว่าเมื่อขยายข้อมูลบล็อคไปทางซ้ายอีกหนึ่งตัวเป็น $x_{i..j+1}$ ย่อมจะเป็นสายอักขระเอกลักษณ์ที่สั้นที่สุดแทนที่ ซึ่งสามารถถูกสร้างสถานะใหม่ขึ้นในแบบจำลองเพื่อรองรับสถานะ $x_{i..j+1}$ และในทำนองเดียวกัน หากพบข้อมูลที่เข้ามาใหม่ $x_{i+k..j+2}$ สำหรับค่า k ใดๆเป็นสายอักขระเอกลักษณ์ที่สั้นที่สุดแล้ว จะมี $x_{i+k-1..j+1}$เป็นคำนำหน้าตัวเดียวเท่านั้น ซึ่งทำให้ $x_{i..j+1}$ เป็นคำนำหน้าตัวเดียวของ $x_{i+k..j+2}$ด้วยเพราะ $x_{i+k-1..j+1}$เป็นคำเดิมท้ายของ $x_{i..j+1}$ทำให้สามารถรวมปลายทางจากสถานะ $x_{i..j+1}$ไปยังสถานะเป้าหมาย $x_{i+k..j+2}$ ได้ แต่หากพบ $x_{i+k..j+2}$ เกิดขึ้นอีกครั้งในตำแหน่งอื่นและมีคำนำหน้าที่ไม่เป็นคำเดิมท้ายของ $x_{i..j+1}$ แล้ว $x_{i+k..j+2}$ ย่อมเป็นสถานะเกิดซ้ำ และสามารถขยายสถานะเอกลักษณ์ใหม่ไปเป็น $x_{i+k-1..j+2}$ ได้ในทำนองเดียวกัน ซึ่งแสดงให้เห็นว่าการปรับสถานะเอกลักษณ์ในแบบจำลองไปเป็นสถานะเกิดซ้ำนั้นสามารถทำได้ตามขั้นตอนวิธีเสมอ □

ทฤษฎีบท 2 การเรียนรู้ด้วยขั้นตอนวิธีตามรูปที่ 3.8 จะมีประสิทธิภาพเชิงเวลาและประสิทธิภาพเชิงพื้นที่ ดีที่สุด $O(n)$ แย่ที่สุด $O(n^2)$ และกรณีเฉลี่ยของประสิทธิภาพเชิงเวลาเท่ากับ $O(n^2)$และประสิทธิภาพเชิงพื้นที่เท่ากับ $O(n)$

พิสูจน์ ให้ข้อมูลอินพุตความยาว n มีสถานะเอกลักษณ์ความยาวเฉลี่ยแต่ละตัวเท่ากับ k ดังนั้นเมื่ออ่านอินพุตแต่ละตัวจะมีสถานะรอทำการ k สถานะ แต่เนื่องจากเฉลี่ยให้ความยาวสถานะเอกลักษณ์เท่ากันทุกตัวดังนั้นจะมีสถานะก่อนหน้าที่เกิดการแตกสถานะใหม่เพียง 1 สถานะ ทำให้เวลาในการดำเนินงานเท่ากับ $k \cdot n$ ซึ่งกรณีที่ k=2 จะทำให้เวลาในการดำเนินงานเท่ากับ $2n$ และ n^2 ในกรณี $k = n$ ในกรณีที่หาค่าเฉลี่ยทำได้โดยพิจารณาค่า k=2,3,...,n จะได้เท่ากับ ค่าเฉลี่ยจากผลรวมเลขคณิตตั้งแต่ 2 ถึง n จะได้ $k = (n^2/2)/n$ซึ่งจะได้ เวลาในการดำเนินงานประมาณ $n^2/2$ จึงสรุปได้ว่าประสิทธิภาพเชิงเวลา $O(n)$ แย่ที่สุด $O(n^2)$ และกรณีเฉลี่ย $O(n^2)$ แต่เมื่อพิจารณาประสิทธิภาพเชิงพื้นที่นั้นไม่ว่า k จะมีค่าเท่าไรก็ตาม ในแต่ละรอบที่อ่านอินพุต n จะมีการสร้างสถานะใหม่เพียง 1 สถานะ และสถานะที่แตกออกเท่ากับ

จำนวนสถานะก่อนหน้าที่มีป้ายสั้นกว่าของสถานะที่แบ่ง ดังนั้นหากพิจารณาเฉลี่ยเมื่อความยาวเท่ากันทั้งหมดแล้ว จะทำให้จำนวนสถานะที่แตกออกใหม่เป็น 1 เท่านั้นจึงมีการสร้างสถานะใหม่จำนวน 2 สถานะในแต่ละรอบ ซึ่งจะได้เท่ากับ $2n$ จะได้ว่าประสิทธิภาพเชิงพื้นที่เฉลี่ยเป็น $O(n)$ และในกรณีที่ดีที่สุด $O(n)$ กรณีที่แย่ที่สุดเป็น $O(n^2)$ □

3.6 การใช้งานเพื่อการจำแนกข้อมูล

การจำแนกข้อมูลในที่นี้หมายถึงปัญหาที่ว่า มีกลุ่มข้อมูล 2 กลุ่มสมมติเป็นกลุ่ม A และกลุ่ม B แล้ว สายอักขระอินพุต S ที่สนใจ นั้นน่าจะเป็นสมาชิกของกลุ่มข้อมูลใดมากกว่ากัน วิธีการแก้ปัญหานี้ด้วยแบบจำลองจากงานวิจัยนี้สามารถทำได้ด้วยการสร้างแบบจำลองขึ้นมา 2 ตัว ออโดมาตาตัวแรกสร้างขึ้นจากข้อมูลกลุ่ม A สมมติแทนได้ด้วย $M(A)$ และออโตมาดาอีกตัวหนึ่งสร้างขึ้นจากข้อมูลกลุ่ม B แทนด้วย $M(B)$ จากนั้นจะใช้ $M(A)$ อ่านอินพุต S เพื่อคำนวณหาความน่าจะเป็นของการเกิดสายอักขระ S แทนด้วย $P(S|M(A))$ และใช้ $M(B)$ อ่านอินพุต S เพื่อคำนวณความน่าจะเป็นของการเกิดสายอักขระ S แทนด้วย $P(S|M(B))$ โดยใช้สมมติฐานที่ว่า "หากว่า S มีองค์ประกอบที่คล้ายคลึ งกับข้อมูลที่สร้างออโดมาดาเชิงน่าจะเป็นแบบสายอักขระย่อยเอกลักษณ์ M_1 มากกว่า M_2 แล้ว แน่นอนว่าความน่าจะเป็ นของ S ที่คำนวณจาก M_1 จะต้องมีค่ามากกว่า M_2" โดยเขียนเป็นสูตรได้ดังนี้

$$l^* = \arg\max_{M} P(M)P(S|M) \qquad 2)$$

จากสมการที่ 2 l^* หมายถึงกลุ่มข้อมูลที่สนใจ คือ ค่า M ที่ทำให้ได้ค่าความน่าจะเป็นตามสมการที่มีค่ามากที่สุด ปัญหาต่อมาคือ จะมีวิธีการคำนวณ $P(S|M(A))$ ได้อย่างไร

3.6.1 วิธีการคำนวณความน่าจะเป็น

กำหนดให้ $S = x_1x_2x_3...x_i$ แทนด้วย $x_{1..i}$ อักขระตำแหน่งที่ i แทนด้วย x_i แล้วความน่าจะเป็นของการเกิด x_i คือ $P(x_i|x_{i-1}, x_{i-2}, ..., x_1)$ ตามหลักการของลูกโซ่มาร์คอฟ แต่เนื่องจากการพิจารณาความน่าจะเป็นด้วยเงื่อนไข x_{i-1} ถึง x_1 สำหรับข้อมูลที่มีขนาดยาวนั้นไม่สามารถทำได้ การคำนวณในที่นี้จึงพิจารณาด้วยเงื่อนไข x_{i-1} ไปถึง x_{i-k} เมื่อ $x_{i-1..i-k}$เป็นสายอักขระเอกลักษณ์ที่ปรากฏบนสถานะของแบบจำลอง ดังนั้น ความน่าจะเป็นในการเกิด xi คือ $P(x_i|Y)$ เมื่อ Y คือป้ายชื่อของสถานะปลายทางที่ได้จากการอ่าน $x_1...x_{i-1}$ โดยที่

$$P(x_i|Y) = \frac{\tau(Y, x_i)}{\sum_{c\in\Sigma}^{n} \tau(Y, c)}$$

เมื่อ $\tau(Y,c)$ คือ ฟังก์ชันความถี่ของการเปลี่ยนสถานะด้วยอักขระ x_i ที่สถานะ Y
ดังนั้นจะสามารถคำนวณความน่าจะเป็นของการเกิดสายอักขระ $S = x_1x_2x_3\ldots x_i$ ใดๆ ได้โดย

$$P(S|M(A)) = P(x_1|\varepsilon)\cdot P(x_2|x_1)\cdot P(x_3|x_2x_1)\cdot\ldots\cdot P(x_i|Y)$$

เมื่อ Y คือสถานะของแบบจำลอง $M(A)$

ตัวอย่าง 3.4

จากรูปที่ 3.7 กำหนดให้ $S = 1010$ แล้ว จะสามารถคำนวณความน่าจะเป็นของการเกิดสายอักขระ 1010 จากแบบจำลองในรูปได้ดังนี้

เริ่มต้น ที่สถานะ $q_{\_\varepsilon}$ อ่าน 1 จะได้ความน่าจะเป็นเท่ากับ 3/8 เปลี่ยนสถานะไป $q_{\_1}$
ที่สถานะ $q_{\_1}$ อ่าน 0 ได้ความน่าจะเป็นเท่ากับ 2/3 เปลี่ยนสถานะไป $q_{\_10}$
ที่สถานะ $q_{\_10}$ อ่าน 1 ได้ความน่าจะเป็นเท่ากับ 1/2 เปลี่ยนสถานะไป $q_{\_101}$
ที่สถานะ $q_{\_101}$ อ่าน 0 ได้ความน่าจะเป็นเท่ากับ 1 เปลี่ยนสถานะไป $q_{\_010}$
จะได้ว่า $P(S|M) =$ 3/8 · 2/3 · 1/2 · 1 = 1/8 หรือเท่ากับ0.125 □

อย่างไรก็ตาม การคำนวณด้วยวิธีการตามที่ได้เห็นในตัวอย่าง 3.4 นั้นอาจจะเกิดปัญหาได้ในกรณีที่เส้นทางการเปลี่ยนสถานะของแบบจำลองไม่มีอยู่จริง ทำให้ความน่าจะเป็นเป็น 0 เช่น จากตัวอย่าง 3.4 หากเปลี่ยนสายอักขระ S เป็น 1011 แล้วจะทำให้สถานะ $q_{\_101}$ อ่าน 1 ได้ความน่าจะเป็นคือ 0 ซึ่งเมื่อนำไปหาความน่าจะเป็นด้วยการคูณความน่าจะเป็นกับตัวอื่นๆ แล้ว ความน่าจะเป็นของสายอักขระ S จะกลายเป็น 0 ไป ทำให้การหาความน่าจะเป็นเกิดข้อผิดพลาดขึ้น คำถามคือ หากไม่ให้ความน่าจะเป็นกลายเป็น 0 ไปเสียแล้ว ควรจะให้ความน่าจะเป็นเป็นเท่าไร การแก้ปัญหาความน่าจะเป็นไม่ให้เป็น 0 นี้จะใช้เทคนิคการปรับเรียบดังหัวข้อ 4.2 ต่อไป

3.6.2 เทคนิคการปรับเรียบ (smoothing technique)

การคำนวณความน่าจะเป็นของสายอักขระ ในกรณีที่เกิดความน่าจะเป็นเมื่อ $P(x_i|x_{i-1}x_{i-2}\ldots x_{i-k})$ มีค่าเป็น 0 มีเทคนิคอย่างหนึ่งที่ใช้กันมากสำหรับการคำนวณหาความน่าจะเป็นสำหรับแบบจำลองภาษาเชิงน่าจะเป็น ได้แก่เทคนิค การปรับเรียบแบบแบ็คอ็อฟ (back off smoothing) ซึ่งจะเป็นการคำนวณความน่าจะเป็นโดยดึงเงื่อนไขอักขระตัวสุดท้ายทางซ้ายออกเพื่อให้ความยาวของเงื่อนไขสั้นลง ซึ่งมีโอกาสจะทำให้ความน่าจะเป็นที่เป็น 0 นั้นมีโอกาสที่จะไม่เป็น 0 โดยทอนตัวทางซ้ายออกให้สั้นลงไปเรื่อยๆจนกว่าจะพบความน่าจะเป็นที่ไม่เป็น 0 เช่น จากตัวอย่าง 3.4 หากคำนวณความน่าจะเป็นของการเกิดสายอักขระ 1011 แล้ว

จะได้ว่า $P(1011|M) = P(1|q_{\_\varepsilon}) \cdot P(0|q_{\_1}) \cdot P(1|q_{\_10}) \cdot P(1|q_{\_101})$ ซึ่งจะได้ 3/8 · 2/3 · 1/2 · 0 จะมีค่าเป็น 0 ดังนั้นตัวที่มีค่าเป็น 0 คือ $P(1|q_{\_101})$ ให้ทำการแบ็คอ็อฟเป็น $P(1|q_{\_01})$ ซึ่งยังมีค่าเป็น 0 จึงทำการแบ็คอ็อฟต่อไปเป็น $P(1|q_{\_1})$ ยังคงเป็น 0 ให้ทำการแบ็คอ็อฟต่อไปจะได้ $P(1|q_{\_\varepsilon})$ มีค่า 3/8 จะได้ว่า $P(1011|M) =$ 3/8 · 2/3 · 1/2 · 3/8 ซึ่งมีค่าความน่าจะเป็นเท่ากับ 0.046875 จะเห็นได้ว่าหากมีสายอักขระ 2 สายระหว่าง 1010 จากตัวอย่าง 3.4 และ 1011 เปรียบเทียบกันว่า สายอักขระใดจะมีความใกล้เคียงกับข้อมูล 101001 มากกว่ากันก็สามารถตอบได้ว่าเป็นสายอักขระ 1010

การปรับเรียบแบบแบ็คอ็อฟที่ได้กล่าวมานั้นมีความสอดคล้องกับแบบจำลอง PUSFA ตรงที่ว่า การเปลี่ยนสถานะของแบบจำลอง PUSFA นั้นสามารถใช้หลักการของเอ็นเอฟเอ ซึ่งในขณะคำนวณจะมีเส้นทางของสายอักขระซึ่งเป็นคำเติมท้ายของสถานะที่มีป้ายชื่อยาวที่สุดทำงานขนานกันไปซึ่งหากพบว่าสถานะใดพบความน่าจะเป็นเป็น 0 ก็ใช้สถานะรอทำการที่เหลือซึ่งสั้นกว่าเป็นตัวคำนวณแทน โดยใช้สูตรการคำนวณดังนี้

Algorithm 3 Probability and Smoothing Technique

```
function probability(inputString)
{       result = 1.0,probability = 1.0,flag = 0
        add ε to activeStates
        for c in inputString
        {       flag = 0
                for state in activeStates
                {       if τ_M(state,c) > 0
                        {       add δ_M(state,c) to nextActiveStates
                                if flag == 0
                                {       probability = P(c|state)
                                        flag = 1}}}
                result = result · probability}}
        add ε to nextActiveStates
        copy nextActiveStates to activeStates
        return result}
```

รูปที่ 3.10 รหัสเทียมฟังก์ชันคำนวณความน่าจะเป็นและเทคนิคการปรับเรียบ

ตัวอย่าง 3.5

จากรูปที่ 3.7 กำหนดให้ $S = 1011$แล้ว จะสามารถคำนวณความน่าจะเป็นของการเกิดสายอักขระ 1011 ด้วยขั้นตอนวิธีตามรูปที่ 3.10 ได้ดังนี้

สถานะ $q_{\_\varepsilon}$ อ่าน 1 ได้ปลายทางเป็น $\{q_{\_\varepsilon}, q_{\_1}\}$ ความน่าจะเป็น $P(1 \mid q_{\_\varepsilon})$ = 3/8
สถานะ $\{q_{\_\varepsilon}, q_{\_1}\}$ อ่าน 0 ได้ปลายทางเป็น $\{q_{\_\varepsilon}, q_{\_0}, q_{\_10}\}$ ความน่าจะเป็น $P(0 \mid q_{\_1})$= 2/3
สถานะ $\{q_{\_\varepsilon}, q_{\_0}, q_{\_10}\}$ อ่าน 1 ได้ปลายทางเป็ น $\{q_{\_\varepsilon}, q_{\_1}, q_{\_01}, q_{\_101}\}$ คำนวณความน่าจะเป็นได้ $P(1 \mid q_{\_10})$ = 1/2
สถานะ $\{q_{\_\varepsilon}, q_{\_1}, q_{\_01}, q_{\_101}\}$ อ่าน 1 ได้ปลายทางเป็น $\{q_{\_\varepsilon}, q_{\_1}\}$ คำนวณความน่าจะเป็นได้ $P(1 \mid q_{\_\varepsilon})$ = 3/8
โดยจะคูณสะสมความน่าจะเป็นเป็น 3/8 · 2/3 · 1/2 · 3/8 ได้ผลลัพธ์เท่ากับ 0.046875 □

การคำนวณความน่าจะเป็นซึ่งใช้ขั้นตอนวิธีตามรูปที่ 3.10 จะใช้ประสิทธิภาพเชิงเวลาเท่ากับ จำนวนความยาวของสายอักขระอินพุตสมมติเป็น n คูณด้วยขนาดของความยาวสูงสุดของป้ายชื่อสถานะสมมติเป็น k หรือ ความยาวสูงสุดของสายอักขระเอกลักษณ์ที่เป็นไปได้ ซึ่งในกรณีที่แย่ที่สุดคือ ความยาวสูงสุดเท่ากับจำนวนความยาวของอินพุต $k = n$ จะได้ว่าประสิทธิภาพเชิงเวลาคือ $O(n^2)$ และในกรณีที่ดีที่สุดคือค่า k มีค่าเท่ากับ จะได้ $O(n)$ ซึ่งในทางปฏิบัติแล้วจะประเมินค่า k โดยใช้ค่าเฉลี่ย k ซึ่งทำให้สรุปได้ว่าประสิทธิภาพเชิงเวลาดีที่สุดจะได้จะเท่ากับ $O(n)$และแย่ที่สุดคือ $O(n^2)$ และประสิทธิภาพเชิงพื้นที่ดีที่สุดคือ $O(1)$และแย่ที่สุดคือ $O(n)$ เพราะในแต่ละรอบจะใช้หน่วยความจำในการเก็บเพียงสถานะรอทำการจำนวน k ตัวเท่านั้น

3.7 สรุป

ในบทนี้ได้นำเสนอ แบบจำลองในการเรียนรู้แบบใหม่เรียกแบบจำลองนี้ว่า PUSFA ซึ่งสามารถเรียนรู้สายอักขระไม่จำกัดความยาวและจำนวนครั้งในการเรียนรู้ ทำให้สามารถเรียนรู้ส่วนเพิ่มได้อย่างไม่จำกัด ดูขั้นดอนวิธี 3.8 และ 3.9 โดยมีอัตราการเพิ่มของพื้นที่ในการเรียนรู้เฉลี่ยแล้วเป็น $O(n)$ ทำให้ไม่ใช้หน่วยความจำมากเกินไป และใช้เวลาในการเรียนรู้ในกรณีดีที่สุด $O(n)$ แย่ที่สุด $O(n^2)$ ซึ่งแน่ นอนว่าข้อมูลมี ชุดตัวอักษรจำกัด อันจะทำให้สถานะเกิดซ้ำนั้นมีจำนวนที่ซ้ำกันมากขึ้นทำให้ไม่ต้องสร้างสถานะใหม่ขึ้น ทำให้แนวโน้มเมื่อเรียนข้อมูลมากขึ้นเรื่อยๆจะทำให้อัตราการเพิ่มขึ้นของสถานะใหม่ลดลงจนเหลือเพิ่มครั้งละ 1 สถานะเมื่ออ่านอินพุต 1 ตัวอักษร นอกจากนี้งานวิจัยนี้ยังเสนอขั้นดอนวิธีในการคำนวณความน่าจะเป็นและขั้นดอนวิธีในการปรับเรียบ เพื่อให้ค่าความน่าจะเป็นที่เป็น 0 นั้นไม่ทำให้การทำงานสะดุด โดย

จะนำการเรียนรู้และการคำนวณค่าความน่าจะเป็นนี้ไปทดลองเพื่อทดสอบหาความแม่นยำในการจำแนกข้อมูล และทดลองหาอัตราการเติบโตของเวลาและพื้นที่ในการเรียนรู้เมื่อเทียบกับทฤษฎีในบทถัดไป

บทที่ 4
ผลการทดลอง

4.1 บทนำ

การทดลองในงานวิจัยนี้ ได้นำแบบจำลองตามที่เสนอมาสร้างขึ้นด้วยการเขียนโปรแกรมตามขั้นตอนวิธีที่ได้กล่าวไว้ในบทที่ 3 โดยใช้ภาษาโปรแกรมไพธอน (python) เวอร์ชัน 3.4.1 ดำเนินงานบนเครื่องคอมพิวเตอร์ระบบปฏิบัติการแมคโอเอส (macOS Sierra) เวอร์ชัน 10.12.5 ซึ่งใช้หน่วยประมวลผลกลางความเร็ว 1.3 จิกะเฮิรทซ์ แบบ 5 แกนประมวลผล หน่วยความจำหลักแบบ DDR3 ขนาด 4 จิกะไบต์ โดยใช้สายอักขระอินพุตที่ใช้ในการเรียนรู้คือ ข้อมูลดีเอ็นของแบคทีเรียอี.โคไล (E. Coli) ซึ่งประกอบไปด้วยชุดตัวอักษร {A, C, G, T} ประกอบขึ้นเป็นสายอักขระความยาวแบ่งเป็นสายละ 58 ตัวอักษรรวมอักขระพิเศษ $ นำหน้า โดยในการทดลองนี้จะตัดส่วนของอักขระพิเศษ # ซึ่งใช้เติมท้ายออกไปเพราะไม่ได้นำไปใช้ในการค้นหาคำเติมท้าย เพื่อลดทอนจำนวนสถานะ โดยมีตัวอย่างดีเอ็นเอของแบคทีเรียอี.โคไล 2 กลุ่มคือ กลุ่มที่เป็นโปรโมเตอร์ 53 สาย และกลุ่มที่ไม่เป็นโปรโมเตอร์ 53 สายรวม 106 สาย โดยสายละ 58 ตัวอักษร รวมความยาวของสายอักขระตัวอย่างที่กล่าวมาทั้งหมด 6,148 ตัวอักษร เหตุผลที่เลือกข้อมูลดีเอ็นเอของแบคทีเรียเนื่องจาก งานวิจัยนี้ต้องการทราบถึงความแม่นยำในการจำแนกข้อมูลด้วยขั้นตอนวิธีแบบออนไลน์ซึ่งสามารถเรียนรู้ส่วนเพิ่มได้ การใช้ข้อมูลดีเอ็นเอที่มีความยาวไม่มากเกินไปจะทำให้สังเกตุเห็นแนวโน้มของผลการทดลองได้เป็นอย่างดี อีกทั้งข้อมูลที่ใช้ในการทดลองนี้เป็นข้อมูลที่เกิดขึ้นจริงซึ่งได้มีงานวิจัยก่อนหน้า [Xing, Pei, Dong, &Yu, 2008] นำไปทดลองใช้กับการเรียนรู้ด้วยวิธีต่างๆ ซึ่งจะทำให้สามารถนำมาเปรียบเทียบความแม่นยำกับงานวิจัยนี้ได้เป็นอย่างดี การทดลองจะใช้ชุดข้อมูลที่เป็นโปรโมเตอร์กับการทดลองที่ 1-3 และชุดข้อมูลทั้ง 2 แบบกับการทดลองที่ 4 เพื่อทดสอบความแม่นยำ และเปรียบเทียบข้อมูลกับงานวิจัยอื่นในการทดลองที่ 5 โดยมีรายละเอียดดังต่อไปนี้

4.2 การทดลองเพื่อทดสอบความถูกต้องทางทฤษฎี

การทดลองเพื่อทดสอบความถูกต้องทางทฤษฎีนั้นมี 3 ส่วนที่ต้องทำการทดลองเพื่อพิสูจน์ความถูกต้องในเชิงทฤษฎีด้วยผลทางปฏิบัติคือ การทดลองที่ 1 สถานะเอกลักษณ์และสถานะเกิดซ้ำของแบบจำลองที่ได้มานั้นตรงกันกับสายอักขระเอกลักษณ์และสายอักขระเกิดซ้ำที่เกิดขึ้นจริงในข้อมูลหรือไม่ การทดลองที่ 2 คือ จำนวนสถานะของแบบจำลองนั้นมีจำนวนเท่าใดเมื่อเทียบกับความยาวอินพุต และมีอัตราการเติบโตของสถานะเมื่อเทียบกับอินพุตตาม

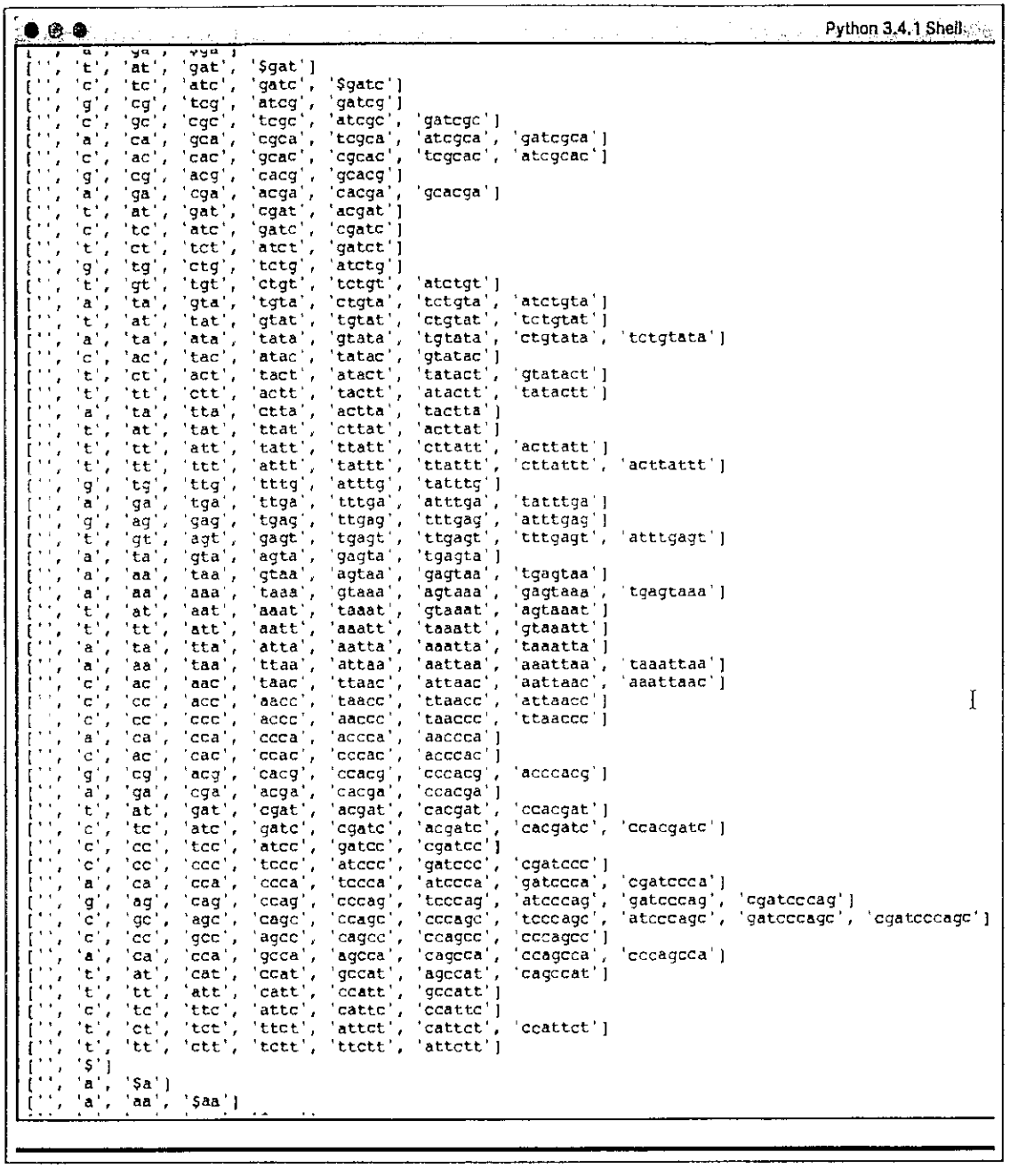
ทฤษฎีหรือไม่ และการทดลองที่ 3 ทดสอบว่าเวลาในการสร้างแบบจำลองนั้นใช้เวลานานเท่าไรมีอัตราการเติบโตเมื่อเทียบกับอินพุตตามทฤษฎีหรือไม่ โดยทำการทดลอง 3 ชุดดังต่อไปนี้

การทดลองที่ 1: ทดสอบความถูกต้องของคุณสมบัติของสถานะเอกลักษณ์และสถานะเกิดซ้ำว่าป้ายชื่อของสถานะเอกลักษณ์และสถานะเกิดซ้ำนั้นตรงตามความเป็นจริงจากข้อมูลหรือไม่ รวมถึงทดสอบว่าสถานะรอทำการนั้นมีคุณสมบัติตรงตามหลักการหรือไม่

วิธีการทดลองที่ 1: ใช้อินพุตดีเอ็นเอของแบคทีเรียกลุ่มที่เป็นโปรโมเดอร์เพื่อสร้างแบบจำลองขึ้นมา โดยใช้ข้อมูลเริ่มจากเพียง 1 สายจนถึง 53 สาย ในการสร้างแบบจำลอง แล้วนำสถานะที่ได้จากแบบจำลองทั้งหมดในแต่ละครั้งแบ่งออกเป็น 2 กลุ่มคือกลุ่มสถานะเอกลักษณ์ และกลุ่มสถานะเกิดซ้ำ จากนั้นจึงทำการเขียนโปรแกรมเพื่อเทียบระหว่างป้ายชื่อสถานะที่ได้จากการเรียนรู้กับข้อมูลอินพุตว่า ป้ายชื่อสถานะเอกลักษณ์และสถานะสายอักขระเกิดซ้ำที่ได้จากแบบจำลองนั้นตรงตามความเป็นจริงและครบถ้วนหรือไม่ กล่าวคือ สถานะเอกลักษณ์จะพบเพียงครั้งเดียวในชุดข้อมูล และสถานะเกิดซ้ำจะพบตั้งแต่ 2 ครั้งขึ้นไปในชุดข้อมูล โดยพบตามจำนวนที่ได้จากฟังก์ชันความถี่ (τ) รวมถึงเขียนโปรแกรมแสดงสถานะที่สร้างขึ้นในขณะดำเนินการแล้วสังเกตุความถูกต้องของสถานะรอทำการในขณะดำเนินงานแต่ละรอบอ่านอินพุต โดยสามารถตรวจสอบความถูกต้องและความสอดคล้องกับทฤษฎีว่า สถานะรอทำการแต่ละสถานะจะต้องเป็นคำเติมท้ายของสถานะรอทำการที่ยาวที่สุด ดูตัวอย่างผลการดำเนินงานได้จากรูปที่ 4.1, 4.2, และ 4.3

รูปที่ 4.1 ตัวอย่างการดำเนินงานโปรแกรมทดสอบจำนวนสถานะเอกลักษณ์เทียบกับอินพุต

รูปที่ 4.2 ตัวอย่างการดำเนินงานโปรแกรมทดสอบจำนวนสถานะเกิดซ้ำเทียบกับอินพุต



รูปที่ 4.3 ตัวอย่างการดำเนินงานโปรแกรมขณะแสดงสถานะรอทำการ

ผลการทดลองที่ 1: ผลการทดลองพบว่า สถานะเอกลักษณ์ทุกสถานะที่ได้จากการดำเนินงานนั้นมีความเป็นเอกลักษณ์ซึ่งพบบนอินพุตเพียงครั้งเดียว ดูตัวอย่างผลการดำเนินงานจากการเขียนโปรแกรมในรูปที่ 4.1 และสถานะเกิดซ้ำทุกสถานะนั้นพบว่ามีในอินพุตจำนวนตั้งแต่ 2 ครั้งขึ้นไปครบทุกสถานะ ดูตัวอย่างผลการดำเนินงานจากการเขียนโปรแกรมในรูปที่ 4.2 อย่างไรก็ตามการทดลองได้ทดลองในกรณีเรียนรู้ส่วนเพิ่มจากข้อมูล 1 สายไปจนถึง 53 สาย ซึ่งทำให้ความยาวอินพุตรวมแล้วเท่ากับ 3074 ตัวอักษร เพื่อให้เกิดสถานะการณ์ที่แตกต่างหลากหลายครบทุกเงื่อนไข โดยทำการสังเกตุผลการทดลองไปจนครบทุกชุดพบว่าได้สถานะเอกลักษณ์ทั้งหมด 2883 สถานะและสถานะเกิดซ้ำจำนวน 3786 สถานะ ซึ่งแต่ละรอบที่เรียนรู้ส่วนเพิ่มที่มากขึ้นนั้นพบว่ายังให้ผลที่ถูกต้องตามทฤษฎี และพบสายอักขระเอกลักษณ์ที่ยาวที่สุดที่มีความยาวถึง 45 ตัวอักษร และสายอักขระเกิดซ้ำที่ยาวที่สุดมีความยาว 44 ตัวอักษร และสังเกตุผลดำเนินงานให้แสดงสถานะรอทำการเมื่อได้อ่านอินพุตแต่ละตัว พบว่าทำงานได้ตรงตามหลักการกล่าวคือ สถานะรอทำการจะเป็นคำเติมท้ายทั้งหมดของสถานะเอกลักษณ์ ดูตัวอย่างการดำเนินงานโปรแกรมตามรูปที่ 4.3

วิเคราะห์ผลการทดลองที่ 1 ผลการทดลองที่ 1 ได้แสดงให้เห็นว่าสถานะเอกลักษณ์และสถานะเกิดซ้ำที่ได้นั้นมีคุณสมบัติตรงตามหลักการ ถูกต้องครบถ้วนทุกสถานะ แสดงให้เห็นว่าการสร้างแบบจำลองนี้ตั้งอยู่บนหลักการที่ถูกต้อง อีกทั้งเซตของสถานะรอทำการแต่ละรอบที่อ่านอินพุตนั้นตรงตามหลักการ แสดงให้เห็นว่าในทางปฏิบัตินั้นได้ผลถูกต้องสอดคล้องกับทางทฤษฎี และที่สำคัญคือ แบบจำลองสามารถจับสายอักขระเอกลักษณ์ที่มีความยาวสูงสุดได้ตรงกับความจริง แสดงให้เห็นว่าแบบจำลองจะสามารถนำไปใช้กับการขึ้นอยู่กับระยะยาวของข้อมูลได้เป็นอย่างดี

การทดลองที่ 2: ทดลองเขียนโปรแกรมเพื่อนับจำนวนสถานะของแบบจำลองที่สร้างขึ้นในแต่ละช่วงที่รับข้อมูลส่วนเพิ่มมาเรียนรู้ เพื่อทดสอบว่าเมื่อข้อมูลที่เรียนรู้มีจำนวนเพิ่มขึ้นนั้นแบบจำลองจะสร้างสถานะขึ้นใหม่ซึ่งจะมีจำนวนสถานะอยู่ในขอบเขตประสิทธิภาพเชิงพื้นที่ตามทฤษฎีหรือไม่

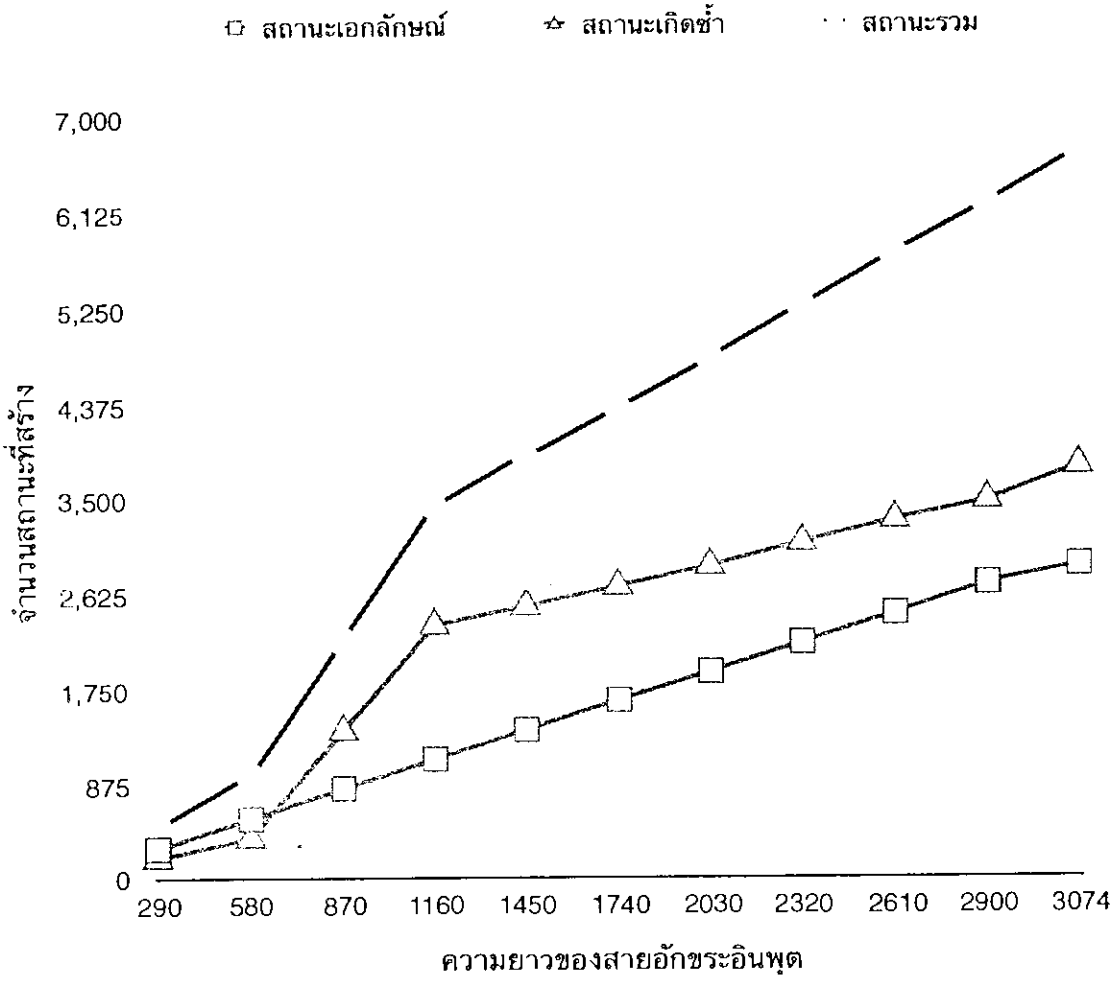
วิธีการทดลองที่ 2: การทดลองทำโดย ใช้อินพุตเป็นข้อมูลดี เอ็นสายละ 58 ตัวอักษร (เพิ่มอักขระ $ นำหน้าต่อกับสายอักขระดีเอ็นเอที่ใช้สำหรับเรียนรู้) ทั้งหมด 53 สาย โดยเริ่มต้นสร้างแบบจำลองจากอินพุตดีเอ็นเอ 5 สาย แล้วทำการเพิ่มครั้งละ 5 สาย (290 ตัวอักษร) ต่อจากของเดิมไปจนครบ 53 สาย (3074 ตัวอักษร) เพื่อนับจำนวนสถานะที่เปลี่ยนไปตามความยาวของอินพุตที่เพิ่มขึ้นทั้งสถานะเอกลักษณ์และสถานะเกิดซ้ำว่ามีจำนวนเท่าไร

ผลการทดลองที่ 2: ผลการทดลองนับจำนวนสถานะเปรียบเทียบกับความยาวอินพุตที่เพิ่มขึ้นแสดงในตารางที่ 4.1

ตารางที่ 4.1 จำนวนสถานะที่สร้างเทียบกับความยาวสายอักขระอินพุต

ความยาวอินพุต	290	580	870	1160	1450	1740	2030	2320	2610	2900	3074
จำนวนสถานะเอกลักษณ์	282	557	834	1101	1371	1643	1905	2173	2442	2718	2883
จำนวนสถานะเกิดซ้ำ	193	379	1367	2339	2510	2692	2878	3096	3302	3475	3786
จำนวนสถานะรวม	475	936	2201	3440	3881	4335	4783	5269	5744	6193	6669

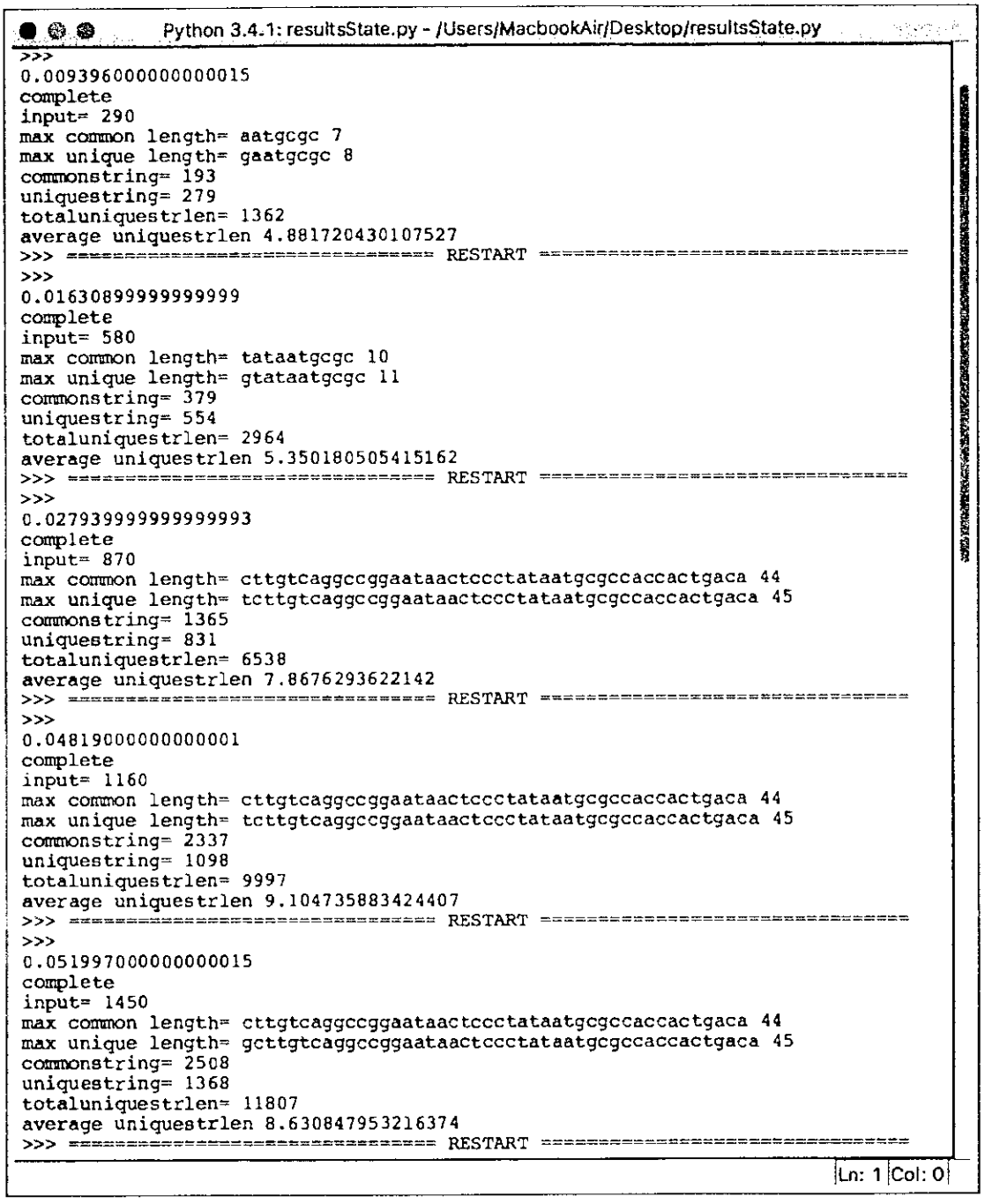
วิเคราะห์ผลการทดลองที่ 2 จำนวนสถานะเอกลักษณ์ตามตารางที่ 4.1 แสดงให้เห็นว่ามีจำนวนเพิ่มขึ้นตามความยาวอินพุต แต่ไม่เกินจำนวนอินพุตเลย และจำนวนสถานะเกิดซ้ำก็มีรูปแบบคล้ายกันคือมีจำนวนเพิ่มขึ้นเมื่อความยาวอินพุตเพิ่มขึ้น แต่จำนวนสถานะเกิดซ้ำนั้นจะมีจำนวนมากกว่าความยาวอินพุต แต่ก็เห็นได้ชัดว่าส่วนใหญ่แล้วไม่เกินจำนวนสองเท่าของความยาวอินพุต แสดงให้เห็นว่าอัตราการเปลี่ยนแปลงของจำนวนสถานะนั้นอยู่ในรูปแบบของสมการเชิงเส้นเมื่อเทียบกับความยาวอินพุต ประมาณ $2n$ ตามที่วิเคราะห์ในทางทฤษฎีในบทก่อนหน้า ดูรูปที่ 4.4 เพื่อให้เห็นภาพอัตราการเติบโตของจำนวนสถานะเมื่อเทียบกับอินพุต โดยแสดงให้เห็นว่าแนวโน้มของจำนวนสถานะเอกลักษณ์ สถานะเกิดซ้ำ และสถานะทั้งสองแบบนั้นเติบโตแบบเส้นตรงเมื่อเทียบกับความยาวของอินพุต และเส้นกราฟสถานะเกิดซ้ำมีอัตราลดลงเมื่อข้อมูลมากขึ้น แสดงว่ามีสถานะเกิดซ้ำที่ซ้ำกันมากขึ้นจึงทำให้สถานะเกิดซ้ำลดลงตามสัดส่วนของข้อมูล ทำให้เห็นว่าผลการทดลองตรงกันกับการวิเคราะห์ทางทฤษฎีที่ว่าอัตราการเติบโตของพื้นที่แย่ที่สุดไม่เกิน $O(n^2)$ และในกรณีเฉลี่ยและดีที่สุด $O(n)$



รูปที่ 4.4 กราฟเปรียบเทียบจำนวนสถานะและความยาวอินพุต

การทดลองที่ 3 ทดลองเขียนโปรแกรมเพื่อคำนวณเวลาในการสร้างแบบจำลองใช้เวลานานเท่าไรเมื่อเทียบกับความยาวอินพุต เมื่อเพิ่มจำนวนข้อมูลมากขึ้นตั้งแต่ 1 สายไปจนถึง 53 สาย

วิธีการทดลองที่ 3 การทดลองวัดเวลาในการประมวลผลตั้งแต่อ่านอินพุตตัวแรกจนถึงตัวสุดท้าย สามารถทำได้ด้วยการบันทึกเวลาในขณะเริ่มทำงาน และเมื่อสิ้นสุดอินพุตตัวสุดท้ายก่อนจบโปรแกรมให้บันทึกเวลาสุดท้ายลบด้วยเวลาเริ่มทำงานจะได้เวลาที่ใช้ไปทั้งหมดในการดำเนินงาน ซึ่งการทดลองจะตัดกระบวนการอื่นๆให้เหลือแต่การสร้างแบบจำลองเพียงอย่างเดียว โดยการทดลองจะใช้อินพุตเช่นเดียวกันกับการทดลองที่ 2 โดยเริ่มจากข้อมูล 5 สายแล้วเพิ่มที่ละ 5 สายไปจนถึงข้อมูลกลุ่มสุดท้าย ทำการจับเวลาในการดำเนินงานแต่ละครั้ง ดูตัวอย่างการดำเนินงานโปรแกรมทดลองที่ 3 ในรูปที่ 4.5



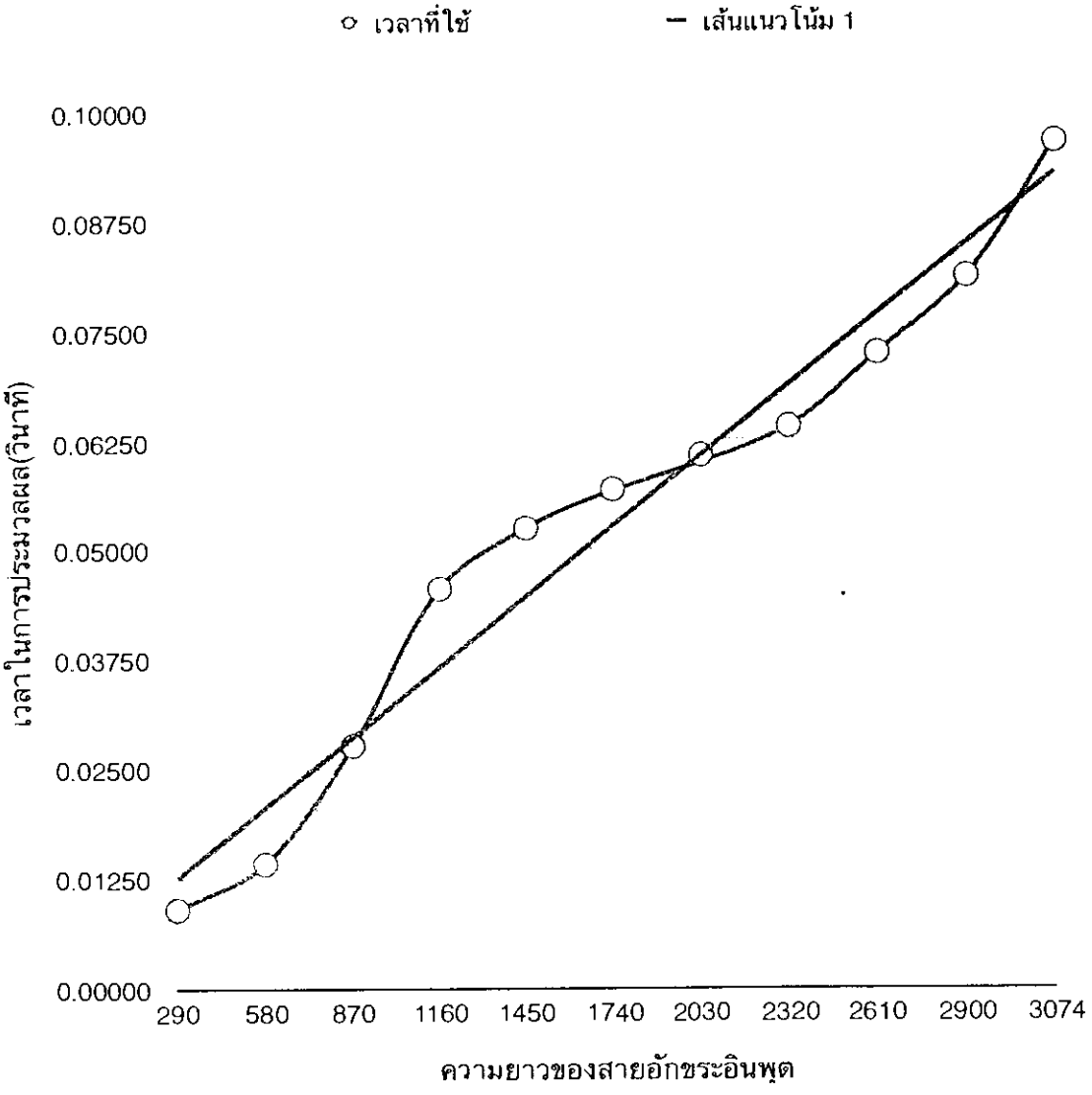
รูปที่ 4.5 ตัวอย่างการดำเนินงานโปรแกรมหาเวลาในการประมวลผล

ผลการทดลองที่ 3 จากรูปที่ 4.5 ตัวเลขตัวแรกในแต่ละครั้งที่ทำการเริ่มเรียนรู้ใหม่คือ เวลาที่ใช้ในการประมวลผล และจำนวนความยาวอินพุตข้อมูลที่ใช้ในการเรียนรู้ (inputs) ความยาวสูงสุดของสายอักขระเกิดซ้ำ (repeating string) และความยาวสูงสุดของสายอักขระเอกลักษณ์ (unique string) และค่าเฉลี่ยความยาวของสายอักขระเอกลักษณ์ที่ได้ในแต่ละรอบการทดลองนับเวลาได้ผลการทดลองตามตารางที่ 4.2

ตารางที่ 4.2 เวลาที่ใช้ในการสร้างแบบจำลองเทียบกับความยาวสายอักขระอินพุต

ความยาวสายอักขระ (อักษร)	290	580	870	1160	1450	1740	2030	2320	2610	2900	3074
เวลาที่ใช้ (วินาที)	0.0093	0.0163	0.0279	0.0481	0.0519	0.0569	0.0608	0.0641	0.0724	0.0811	0.0965

วิเคราะห์ผลการทดลองที่ 3 เวลาในการประมวลผลใช้เวลารวดเร็วเมื่อเทียบกับอินพุตความยาว 3074 ตัวอักษรใช้เวลาเพียงไม่ถึง 0.1 วินาที และอัตราการเพิ่มขึ้นของเวลาในการประมวลผลนั้นเพิ่มขึ้นในสัดส่วนค่อนข้างคงที่ในแนวเส้นตรงเมื่อเทียบกับความยาวของอินพุต ซึ่งแสดงให้เห็นว่าประสิทธิภาพเชิงเวลาในการสร้างแบบจำลองในงานวิจัยนี้ในทางปฏิบัติแล้วบางช่วงโค้ง บางช่วงตรง ซึ่งแสดงให้เห็นว่าบางช่วงใช้เวลามากแบบยกกำลัง บางช่วงใช้เวลาน้อยแบบเส้นตรง แต่เมื่อดูแนวโน้มทั้งหมดแล้วเป็นการเติบโตเฉลี่ยแบบเชิงเส้น ซึ่งสอดคล้องกับทฤษฎีที่ว่าอัตราการเติบโตของเวลาในการประมวลผลไม่เกิน $O(n^2)$ และทำได้ดีที่สุด $O(n)$ โดยดูกราฟการเติบโตของเวลาในการประมวลผลได้ในรูปที่ 4.6 ซึ่งประมาณการได้ว่าเป็นเส้นตรงที่มีความชันประมาณ $3.14*10^{-5}$ วินาที/ความยาว 1 ตัวอักษร โดยมีแกน x เป็นความยาวอินพุตและมีแกน y เป็นเวลาในการประมวลผล โดยสามารถประมาณการได้ว่าในกรณีที่อัตราการดำเนินงานคงที่เช่นนี้ ในเชิงปฏิบัติหากนำไปใช้กับอินพุตความยาว 1 ล้านตัวอักษรด้วยอัตราเดียวกันนี้ก็อาจจะทำเวลาได้ดีประมาณ 31.4 วินาที



รูปที่ 4.6 กราฟแสดงเวลาที่ใช้ในการดำเนินงานเมื่อเทียบกับความยาวของอินพุต

4.3 การทดลองเพื่อทดสอบความแม่นยำในการจำแนกข้อมูล

ในหัวข้อนี้จะแบ่งการทดลองเป็น 2 เรื่องคือการทดลองที่ 4 จะทดลองเพื่อทดสอบความแม่นยำในการจำแนกข้อมูล และในการทดลองที่ 5 จะทดลองเปรียบเทียบความแม่นยำกับวิธีการอื่นๆ โดยการจำแนกข้อมูลนั้นสมมติว่า ในสถานการณ์ที่ว่ามีข้อมูลกลุ่ม A และมีข้อมูลกลุ่ม B และมีข้อมูล S ที่ไม่เคยพบมาก่อน การจำแนกข้อมูลคือการระบุได้ว่าข้ อมูล S จะมีลักษณะที่ควรเป็นสมาชิกในกลุ่ม A หรือ B ในงานวิจัยนี้จะทดสอบความแม่นยำในการจำแนกข้อมูลด้วยแบบจำลองในงานวิจัยนี้ โดยจะใช้แบบจำลองในงานวิจัยนี้ 2 ตัวที่สร้างขึ้นจากข้อมูล

กลุ่ม A และอีกตัวหนึ่งสร้างขึ้นจากข้อมูลกลุ่ม B (training data) และจะต้องมีกลุ่มข้อมูลอินพุตเพื่อทดสอบ (testing data)โดยให้เป็นสายอักขระอินพุตให้กับเครื่องจักรทั้ง 2 ตัวเพื่อคำนวณว่าตัวใดมีความน่าจะเป็นสูงกว่ากัน โดยกลุ่มข้อมูลอินพุตที่จะทดสอบนั้นจะต้องไม่เหมือนกันกับข้อมูลที่ใช้สร้างแบบจำลอง หากว่านำอินพุตทดสอบมาคำนวณด้วยเครื่องจักร $M(A)$ มีความน่าจะเป็นสูงกว่าก็แสดงว่า อินพุตนั้นควรจะเป็นสมาชิกของ $M(A)$ ซึ่งมีประเด็นที่น่าสนใจคือ

1. ความแม่นยำในการจำแนกข้อมูลคิดเป็นร้อยละเท่าไร และ
2. แบบจำลองในงานวิจัยนี้มีประสิทธิภาพในการจำแนกข้อมูลเป็นอย่างไรเมื่อเทียบกับงานวิจัยอื่นซึ่งเป็นที่รู้จักดีและมีประสิทธิภาพได้รับการยอมรับในวงกว้าง

จากทั้งสองประเด็นนี้นำไปสู่การทดลองเพื่อหาข้อสรุปโดยมีรายละเอียดดังนี้

การทดลองที่ 4 ทดสอบความแม่นยำในการจำแนกข้อมูล โดยวัดจำนวนครั้งที่แบบจำลองสามารถจำแนกกลุ่มได้ถูกต้องและจำนวนครั้งที่แบบจำลองจำแนกผิดไปจากความจริง

วิธีการทดลองที่ 4 การทดลองเพื่อทดสอบความแม่นยำในการจำแนกข้อมูลนั้น เพื่อประโยชน์ในการเปรียบเทียบประสิทธิภาพกับงานวิจัยอื่นๆภายใต้สถานการณ์และข้อมูลชุดเดียวกันกับงานวิจัยที่เกี่ยวข้อง ในงานวิจัยนี้ได้เลือกใช้ข้อมูลดีเอ็นเอของแบคทีเรียอี.โคไล โดยแบ่งกลุ่มข้อมูลดีเอ็นเอจากชุดข้อมูลตัวอย่าง 2 กลุ่ม กลุ่มแรกเป็นข้อมูลดีเอ็นเอที่เป็นโปรโมเตอร์ (promoters) และอีกกลุ่มหนึ่งเป็นดีเอ็นเอที่ไม่เป็นโปรโมเตอร์ (non-promoters) โดยโปรโมเตอร์จะเป็นส่วนหนึ่งของดีเอ็นเอที่อยู่ส่วนต้นของยีน มีรูปแบบการจัดเรียงนิวคลีโอไทด์เฉพาะตัว ซึ่งถูกนำไปใช้ประโยชน์ทางการวิเคราะห์ดีเอ็นเอ การทดลองนี้จะนำข้อมูลดีเอ็นเอทั้งสองกลุ่มมาสร้างแบบจำลองเพื่อรู้จำทั้งส่วนที่เป็นโปรโมเตอร์และส่วนที่ไม่เป็นโปรโมเตอร์ แล้วนำแบบจำลองที่ได้นั้นไปทดสอบจำแนกส่วนของดีเอ็นเอที่ไม่เคยพบมาก่อนว่าจะเป็นโปรโมเตอร์หรือไม่

ข้อมูลดีเอ็นเอของแบคทีเรียจะมี 2 กลุ่มคือกลุ่มโปรโมเตอร์ และกลุ่มที่ไม่เป็นโปรโมเตอร์ จำนวนอย่างละ 53 สาย สายละ 57 ความยาวตัวอักษร รวมแล้วจำนวน 106 สาย การแบ่งข้อมูลเพื่อใช้สำหรับทดสอบนั้นจะใช้หลักการเอาออกหนึ่งสาย (leave one out) เพื่อนำข้อมูลที่เอาออกจากกลุ่มที่ใช้สร้างแบบจำลอง ให้เป็นตัวทดสอบที่แบบจำลองไม่เคยพบมาก่อน เช่น ข้อมูลโปรโมเตอร์ 53 สาย เอาออก 1 สายจะเหลือข้อมูลที่นำไปสร้างแบบจำลอง 52 สาย โดยทำซ้ำจำนวน 53 ครั้ง และอีกกลุ่มหนึ่งคือชุดที่ไม่เป็นโปรโมเตอร์ก็ทำการสร้างแบบจำลองด้วยข้อมูล 53 สาย และทำในวิธีการเช่นเดียวกันนี้แต่เปลี่ยนชุดที่ดึงออกทำเช่นนี้จนครบ 53 สาย โดยใช้วิธีการนี้กับกลุ่มที่ไม่เป็นโปรโมเตอร์ด้วย ข้อมูลแต่ละสายที่ดึงออกมาเพียงสายเดียวในแต่ละครั้งนั้นจะถูกนำมาทดสอบด้วยแบบจำลองว่า มีความน่าจะเป็นเท่าไร และมีความน่าจะเป็นในกลุ่มใดมากกว่ากัน โดยใช้หลักการจำแนกเช่นเดียวกันกับที่เสนอในบทที่ 3

รูปที่ 4.7 ตัวอย่างการดำเนินงานโปรแกรมขณะแสดงการจำแนกกลุ่มข้อมูล

ผลการทดลองที่ 4

ผลการทดลองได้ดำเนินงานดามรูปที่ 4.7 โดยได้ผลลัพธ์จากการดึงข้อมูลออกจำนวน 106 ตัวแบ่งเป็นกลุ่มโปรโมเดอร์ 53 สายและกลุ่มไม่เป็นโปรโมเดอร์ 53 สายพบว่าแบบจำลองทำนายกลุ่มโปรโมเดอร์ผิดพลาดเพียง 4 สายจาก 53 สาย โดยแบบจำลองได้ทำนายสายดีเอ็นเอที่เป็นโปรโมเดอร์ว่าไม่เป็น ซึ่งถือเป็นผลลบลวง (false negative) และกลุ่มที่ไม่เป็นโปรโมเดอร์นั้นแบบจำลองสามารถจำแนกกลุ่มได้ถูกต้องครบทุกตัว ดังนั้นแบบจำลองที่นำเสนอในงานวิจัยนี้สามารถทำได้ถูกต้องได้ถึง 102 สายจาก 106 สาย ผิดพลาดเพียง 4 สายจาก 106 สาย ซึ่งถูกต้องแม่นยำถึงร้อยละ 96.23

วิเคราะห์ผลการทดลองที่ 4 ความแม่นยำของแบบจำลองในงานวิจัยนี้ทำได้ถึงร้อยละ 96.23 โดยมีความผิดพลาดแบบผลลบลวงจำนวน 4 สายจาก 106 สาย ซึ่งเป็นการบอกตัวที่เป็นโปรโมเตอร์ว่าไม่เป็นโปรโมเตอร์ โดยถือได้ว่าเป็นความผิดพลาดรอง อันเปรียบเทียบได้กับตำรวจปล่อยผู้ร้ายตัวจริงเพราะวินิจฉัยผิดย่อมดีกว่าจับคนผิด และสามารถจำแนกสายอักขระที่ไม่เป็นโปรโมเตอร์ได้ถูกต้องครบถ้วนซึ่งถือได้ว่า ผลบวกลวง (false positive) อันเป็นความผิดพลาดหลักนั้นมีค่าเป็น 0 เปรียบเทียบได้กับตำรวจจับคนดีมาเป็นผู้ร้ายซึ่งเป็นข้อผิดพลาดหลักซึ่งไม่เกิดขึ้นในงานวิจัยนี้ แบบจำลองในงานวิจัยนี้จึงมีประสิทธิภาพสูงมากในการจำแนกกลุ่มข้อมูล

การทดลองที่ 5 เปรียบเทียบประสิทธิภาพในการจำแนกข้อมูลระหว่างแบบจำลองในงานวิจัยนี้และงานวิจัยอื่นที่มีประสิทธิภาพและได้รับการยอมรับในวงกว้าง

วิธีการทดลองที่ 5 การเปรียบเทียบประสิทธิภาพระหว่างแบบจำลองในงานวิจัยนี้กับงานวิจัยอื่นนั้นจำเป็นต้องเปรียบเทียบงานที่มีการทดลองกับชุดทดลองชุดเดียวกัน โดยแบ่งเป็น 2 วิธีคือ 1.เลือกจากงานวิจัยที่มีผลการทดลองมาแล้ว และ 2. เขียนโปรแกรมเพิ่มเติมเพื่อทดลองกับวิธีการที่ยังไม่มีผลทดลองในชุดทดลองเดียวกัน ซึ่งจากการสำรวจงานวิชาการพบว่างานวิจัยที่ได้ทำการทดลองกับชุดทดลองที่เหมือนกันนี้ได้ใช้วิธีการที่ได้รับการยอมรับในวงกว้างปรากฏผลแล้วในงานวิจัยก่อนหน้า [Xing, Pei, Dong, &Yu, 2008] และ [Towell, Shavlik, & Noordewier, 1990] คือ

1. วิธีการโครงข่ายประสาทเทียมแบบฐานความรู้ (knowledge based neural network)หรือ KBANN

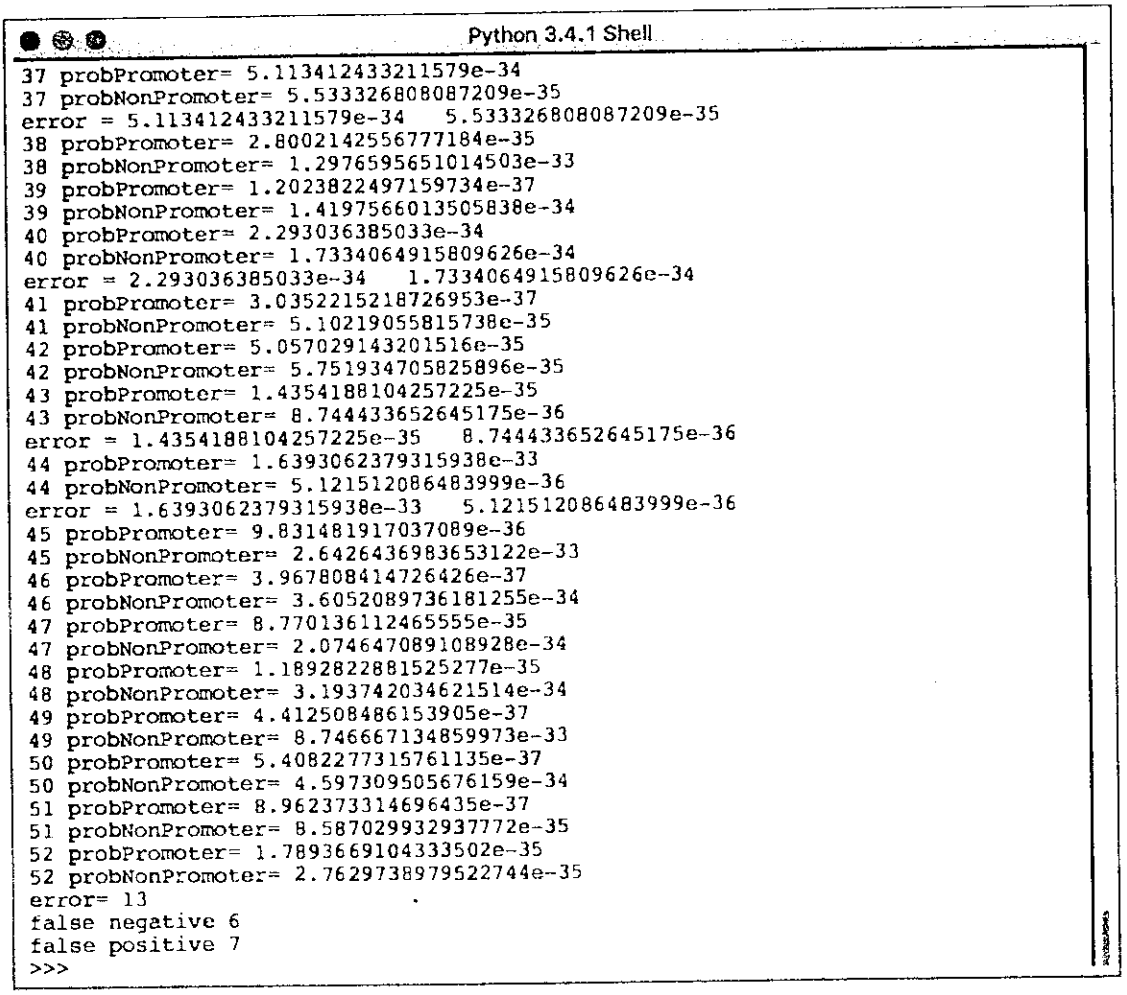
2. วิธีการโครงข่ายประสาทเทียมมาตรฐาน (standard neural network) แบบส่งค่าย้อนกลับ (back propagation) หรือ BP

3. วิธีการ กฎการจำแนกลำดับ (sequential classification rule:SCR) และ ต้นไม้ตัดสินใจสำหรับลำดับแบบทั่วไป (generalized sequential decision tree: GSDT)

4. วิธีการเฉพาะทางในทางชีวสารสนเทศของโอนีล (O'neil)

5. วิธีการต้นไม้ตัดสินใจ (decision tree) เรียกโดยย่อ ID3

และเพิ่มเติมทดลองด้วยการสร้างโปรแกรมตามวิธีการเอ็นแกรม (n-gram) โดยใช้ความยาว 2, 3, 4 และ 8 ซึ่งเป็นความยาวเฉลี่ยของสายอักขระเอกลักษณ์ของชุดข้อมูลตามที่ปรากฏในงานวิจัยนี้ ดูตัวอย่างการดำเนินงาน 3-gram ในรูปที่ 4.7



รูปที่ 4.8 ตัวอย่างการดำเนินงานโปรแกรมจำแนกกลุ่มข้อมูลด้วยวิธีการ 3-gram

ผลการทดลองที่ 5: การทดลองได้นำเอาข้อมูลที่ได้จากงานวิจัยก่อนหน้า และจากการเขียนโปรแกรมด้วยวิธีการเอ็นแกรมโดยใช้ 2, 3, 4, และ 8 แกรม มาเปรียบเทียบความผิดพลาดทั้งผลบวกลวง และผลลบลวง และความผิดพลาดรวมทั้งหมด ได้ผลเปรียบเทียบดังตารางที่ 4.3 โดยมีจำนวนข้อมูลทดสอบทั้งหมด 106 สาย

ตารางที่ 4.3 จำนวนความผิดพลาดในการจำแนกข้อมูลด้วยวิธีการต่างๆ

วิธีการ/ความผิดพลาด	Error(false positive)	Error(false negative)	Total error (from 106)
KBANN	1	3	4
BP	ไม่มีข้อมูล	ไม่มีข้อมูล	8
O'neill	0	12	12
ID3	13	8	21
SCR	ไม่มีข้อมูล	ไม่มีข้อมูล	7
GSDT	ไม่มีข้อมูล	ไม่มีข้อมูล	10
3-NN	9	4	13
2-gram	9	9	18
3-gram	7	6	13
4-gram	5	14	19
8-gram	2	6	8
PUSFA	0	4	4

วิเคราะห์ผลการทดลองที่ 5 จากตารางที่ 4.3 แสดงให้เห็นว่าแบบจำลองที่ได้จากงานวิจัยนี้ (PUSFA) สามารถจำแนกข้อมูลได้แม่นยำที่สุดเทียบเท่ากับวิธีการ KBANN คือผิดพลาดรวมแล้วเพียง 4 สายจากการทดสอบ 106 สาย แต่เมื่อวิเคราะห์ในส่วนของผลบวกลวงแล้ว PUSFA ทำได้ดีกว่าเพราะไม่เกิดผลบวกลวงเลยในขณะที่ KBANN เกิดผลบวกลวง 1 สาย สำหรับวิธีการที่ได้รับความนิยมสูงเช่นเอ็นแกรมนั้น ทำได้ดีที่สุดที่ความยาว 8 เพราะเป็นความยาวที่ได้จากการคำนวณค่าเฉลี่ยจากแบบจำลอง PUSFA ซึ่งเกิดความผิดพลาด 8 สาย ซึ่งแสดงให้เห็นว่า การเรียนรู้ด้วยสายอักขระเอกลักษณ์นั้นเป็นการเรียนรู้ที่รวดเร็วและแม่นยำมากตัวหนึ่ง

บทที่ 5
สรุปผลการวิจัยและข้อเสนอแนะ

5.1 สรุปผลการวิจัย

งานวิจัยนี้ได้เสนอแบบจำลองแบบใหม่และขั้นตอนการเรียนรู้จากสายอักขระเพื่อสร้างแบบจำลอง ซึ่งสามารถใช้แบบจำลองนี้สำหรับเก็บข้อมูลเพื่อใช้ค้นหาตัวอย่างสายอักขระย่อยที่เกิดในชุดข้อมูล รวมถึงสายอักขระย่อยเอกลักษณ์ และสายอักขระเกิดซ้ำ และประเด็นสำคัญคือเพื่อใช้ในการจำแนกสายอักขระซึ่งเป็นลำดับเชิงสัญลักษณ์ โดยคำนึงถึงข้อได้เปรียบคือ แบบจำลองจะต้องสามารถเรียนรู้เพิ่มเติมจากข้อมูลใหม่ได้ทันทีได้โดยไม่ต้องใช้ข้อมูลเก่ารวบรวมกลับมาเรียนรู้ในครั้งใหม่ ทำให้แบบจำลองสามารถรองรับการเรียนรู้ข้อมูลจำนวนมากและมีความยาวต่อเนื่องได้ รวมถึงสามารถเรียนรู้ได้ไม่จำกัดจำนวนครั้ง เพื่อให้มีศักยภาพรองรับข้อมูลที่มีปริมาณมากในปัจจุบัน ซึ่งในการเรียนรู้นั้นจะต้องมีประสิทธิภาพทางเวลาและพื้นที่ที่ดีเพียงพอต่อการรองรับข้อมูลปริมาณมาก และไม่ใช่เพียงแต่การรองรับการเรียนรู้ส่วนเพิ่มเท่านั้น แบบจำลองจะต้องมีความแม่นยำในการจำแนกข้อมูลได้ โดยตั้งเป้าหมายไว้สำหรับการนำไปใช้งานเกี่ยวกับสายอักขระข้อมูลดีเอ็นเอที่มีจำนวนมหาศาลในปัจจุบันเพื่อที่จะเป็นส่วนหนึ่งในการนำไปพัฒนาสำหรับการรักษาโรคและพัฒนาคุณภาพชีวิตโดยการถอดรหัสดีเอ็นเอ อย่างไรก็ตามหลักการที่สำคัญของการสร้างเครื่องมือสำหรับเรียนรู้นั้นจำกัดไว้ว่าประสิทธิภาพเชิงเวลาและพื้นที่ไม่ควรเกินเวลาเชิงพหุนาม ดังนั้นแบบจำลองที่นำเสนอนี้จึงค้นคว้าทดลองสังเกตจนได้หลักการที่ลงตัวสำหรับสร้างแบบจำลองเพื่อใช้ในการเรียนรู้ภายใต้ข้อจำกัดและจุดประสงค์ตามที่กล่าวมา โดยใช้หลักการจากสายอักขระเอกลักษณ์เพื่อจำกัดความยาวของการหาคำนำหน้าที่เป็นไปได้ทั้งหมด เพื่อลดจำนวนสถานะที่อาจสร้างขึ้นอย่างไม่มีขอบเขตตามจำนวนความยาวอินพุต

หลักการในการเรียนรู้นั้นเพื่อให้ได้แบบจำลองที่มีคุณสมบัติหลักของสถานะคือ สถานะจะมี 2 แบบคือสถานะเอกลักษณ์และสถานะเกิดซ้ำ ซึ่งเป็นสถานะที่ใช้แทนข้อมูลสายอักขระเอกลักษณ์และสายอักขระเกิดซ้ำตามลำดับ ซึ่งขั้นตอนวิธีในการสร้างแบบจำลองนี้สามารถดำเนินการได้ในรูปแบบเชื่อมตรง (on-line algorithm) ทำให้สามารถเรียนรู้ส่วนเพิ่มได้ไม่จำกัด และด้วยหลักการสถานะเอกลักษณ์ที่สามารถเปลี่ยนสภาพไปเป็นสถานะเกิดซ้ำจะทำให้สามารถปรับโครงสร้างแบบจำลองจากการเรียนรู้ส่วนเพิ่มได้โดยไม่ต้องรวบรวมข้อมูลเก่ามาเรียนรู้ร่วมกับข้อมูลใหม่ ซึ่งพิสูจน์ให้เห็นทั้งในทางทฤษฎีและปฏิบัติการทดลองว่ามีการทำงานที่ถูกต้องตามหลักการ ซึ่งหมายความว่า การเรียนรู้จากสายอักขระ ABC ในครั้งเดียว ให้ผลลัพธ์เหมือนกันกับการเรียนรู้จาก A แล้วเพิ่ม B และเพิ่ม C ต่อจากของเดิม สามารถดูได้จากการทดลองที่

2 ในบทที่ 4 จึงเป็นเครื่องยืนยันว่าแบบจำลองนี้จึงสามารถเรียนรู้ส่วนเพิ่มได้อย่างถูกต้อง อย่างเป็นรูปธรรม

ประเด็นต่อมาคือ ความเร็วในการเรียนรู้นั้นในเชิงทฤษฎีจะขึ้นอยู่กับความยาวของสายอักขระเอกลักษณ์ที่มีอยู่ในข้อมูล แต่อย่างไรก็ดีเวลาในการเรียนรู้ที่ทำได้ดีที่สุดอยู่ในรูปฟังก์ชันเชิงเส้นของความยาวอินพุด $O(n)$ และแย่ที่สุดไม่เกินฟังก์ชันยกกำลังสองของความยาวอินพุด $O(n^2)$ ซึ่ งแสดงให้เห็นจากการทดลองที่ 3 พบว่ากราฟของเวลาที่ใช้ในการประมวลผลเดิบโดแบบเชิงเส้น และทำเวลาในทางปฏิบัติได้อย่างรวดเร็วโดยประเมินอย่างคร่าวๆว่าหากข้อมูลมีความยาว 1 ล้านตัวอักษรแล้วก็อาจจะใช้เวลาประมาณ 32 วินาที ซึ่งข้อมูลดีเอ็นเอของมนุษย์นั้นมีอยู่หลัก 1000 ล้านตัวอักษร หากนำมาประมวลผลทั้งหมดก็จะใช้เวลาประมาณ 9 ชั่วโมง ซึ่งแบบจำลองนี้จะทำให้การดำเนินการเกี่ยวกับข้อมูลปริมาณมหาศาลอยู่ในขอบเขตที่จัดการได้

สิ่งที่พิจารณาต่อไปอีกคือ ประสิทธิภาพเชิงพื้นที่ของการเรียนรู้เมื่อเทียบกับความยาวอินพุดนั้น พบว่าเนื่องจากพื้นที่ที่ใช้จัดเก็บข้อมูลคือ จำนวนสถานะของแบบจำลองซึ่งแบ่งออกเป็น 2 ประเภทคือ สถานะเอกลักษณ์และสถานะเกิดซ้ำ สถานะเอกลักษณ์นั้นมีจำนวนไม่เกินความยาวอินพุดเพราะหากเลื่อนบล็อคสายอักขระเอกลักษณ์ไปทีละตัวอักษรดลอดทั้งอินพุดก็จะได้จำนวนไม่เกินความยาวอินพุด และสถานะเกิดซ้ำนั้นเป็นสถานะที่เก็บสายอักขระย่อยซึ่งเป็นคำเติมท้ายของสายอักขระเอกลักษณ์ทั้งหมด ซึ่งขึ้นอยู่กับว่าสถานะเอกลักษณ์มีความยาวเฉลี่ยเท่าไร โดยการวิเคราะห์ในบทที่ 3 สถานะรวมทั้งสองประเภทจะมีจำนวนน้อยที่สุดเดิบโดแบบฟังก์ชันเชิงเส้นตามจำนวนอินพุด $O(n)$ และมีจำนวนมากที่สุดเดิบโดไม่เกินฟังก์ชันกำลังสองของอินพุด $O(n^2)$ ซึ่งในการทดลองให้ผลการทดลองเช่นเดียวกันกับทฤษฎีในรูปแบบที่ดีที่สุดคือ จำนวนสถานะเดิบโดแบบเชิงเส้น ซึ่งแสดงให้เห็นว่าการเรียนรู้ด้วยแบบจำลองจะสามารถจัดการได้โดยใช้หน่วยความจำที่ไม่มากเกินไปนักเมื่อเทียบการจัดเก็บข้อมูลอินพุดตามปกติ

เมื่อได้แบบจำลองจากการเรียนรู้อินพุดมาแล้วนั้น สิ่งสำคัญคือการนำแบบจำลองไปใช้งาน เพื่อการจำแนกกลุ่มข้อมูล ประเด็นความแม่นยำในการจำแนกข้อมูล จึงต้องนำมาพิสูจน์ว่ามีความแม่นยำมากน้อยเพียงใด และเทียบกับงานวิจัยอื่นๆแล้วนั้นได้ผลอย่างไร ซึ่งหากพิจารณาจากทฤษฎีแล้วจะพบว่า สายอักขระเอกลักษณ์คือสายอักขระที่เกิดขึ้นเพียงครั้งเดียว ดังนั้นความน่าจะเป็นจะต้องเป็น 1 ในการเกิดอักขระตัวต่อไป ซึ่งสายอักขระเอกลักษณ์มีความยาวมากและตรงกันกับอินพุดมากเท่าไรยิ่งทำให้ความน่าจะเป็นนั้นมีค่าสูงกว่าการใช้วิธีการอื่นๆ ซึ่งบ่งบอกความแม่นยำในจำแนกข้อมูล อีกประเด็นสำคัญคือ การจดจำรูปแบบสายอักขระเอกลักษณ์นั้นเหมาะสมกับข้อมูลอินพุดที่มีรูปแบบเฉพาะตัว ดังเช่นข้อมูลดีเอ็นเอที่เป็นโปรโมเดอร์ในการทดลองในบทที่ 4 จะมีรูปแบบเฉพาะตัวไม่เหมือนกับข้อมูลที่ไม่เป็นโปรโมเดอร์ จึงทำให้ผลการทดลองในการจำแนกข้อมูลทำได้ดีที่สุดเมื่อเทียบกับงานวิจัยอื่น ซึ่ง

มีความถูกต้องแม่นยำ 102 ตัวอย่างจากทั้งหมด 106 ตัวอย่างคิดเป็นร้อยละ 96.23 ซึ่งสูงสุดเมื่อเทียบกับวิธีการอื่น โดยเทียบเท่ากับงานวิจัยที่ใช้วิธีการโครงข่ายประสาทเทียมแบบผสมผสาน ซึ่งผิดพลาด 4 ครั้งเท่ากัน ซึ่งในนั้นเป็นผลบวกลวง 1 ครั้ง ที่สำคัญคือ วิธีการที่ได้จากงานวิจัยนี้เป็นการผิดพลาดแบบรองซึ่งเป็นผลลบลวงทั้งหมด ไม่มีผลบวกลวงเลย แบบจำลองในงานวิจัยนี้จึงมีความเหมาะสมและมีศักยภาพอย่างยิ่งในการเรียนรู้เพื่อจำแนกข้อมูลสายอักขระที่มีรูปแบบดังเช่นข้อมูลดีเอ็นเอที่มีการคัดลอกยีนและโครโมโซมซึ่งมีเอกลักษณ์เฉพาะตัว

ประเด็นสุดท้ายคือ ความเร็วในการจำแนกข้อมูลของแบบจำลองในงานวิจัยนี้ มีข้อได้เปรียบมากกว่างานวิจัยอื่นๆคือ แบบจำลองนี้ปรับปรุงมาจากแบบจำลองออโดมาดาซึ่งมีศักยภาพการทำงานแบบเวลาจริง กล่าวคือ สามารถอ่านอินพุตหนึ่งตัวแล้วดำเนินการได้ทันทีโดยใช้วิธีการเปลี่ยนสถานะไปตามขั้น ทำให้การจำแนกข้อมูลด้วยวิธีการในงานวิจัยนี้มีความรวดเร็วเมื่อเทียบกับวิธีการเรียนรู้ซึ่งเป็นที่นิยมอื่นๆ จึงเหมาะสมอย่างยิ่งกับการเรียนรู้อินพุตที่จะต้องมีการตัดสินใจแบบเวลาจริง เช่น การป้องกันการบุกรุก ปัญญาประดิษฐ์ของหุ่นยนต์ การควบคุมเครื่องจักร และทางการทหาร

จึงขอสรุปผลการวิจัยว่า แบบจำลองที่ได้จากงานวิจัยนี้ สามารถนำไปใช้เรียนรู้ข้อมูลประเภทลำดับเชิงสัญลักษณ์หรือสายอักขระ โดยสามารถเรียนรู้ในส่วนเพิ่มได้อย่างต่อเนื่อง เพื่อรองรับข้อมูลปริมาณมากในอนาคตได้อย่างสืบเนื่องต่อไป โดยมีศักยภาพในการจำแนกข้อมูลได้อย่างแม่นยำ และมีความรวดเร็วทั้งในการเรียนรู้และการจำแนกข้อมูล และใช้หน่วยความจำสำหรับเรียนรู้ข้อมูลส่วนเพิ่มในอัตราส่วนไม่เกินฟังก์ชันพหุนามโดยพิสูจน์จากทฤษฎีและปฏิบัติได้ผลเป็นที่น่าพอใจ ตรงตามวัตถุประสงค์ที่วางไว้ อันสามารถจะเป็นพื้นฐานสำคัญและเป็นประโยชน์ต่องานวิจัยเกี่ยวกับศาสตร์ทางคอมพิวเตอร์และอื่นๆต่อไป

5.2 ข้อเสนอแนะและงานวิจัยต่อยอด

แบบจำลองในงานวิจัยนี้สามารถเก็บสายอักขระย่อยของอินพุต ทั้งที่เป็นสายอักขระเอกลักษณ์และสายอักขระเกิดซ้ำ ทำให้สามารถนำไปใช้สำหรับปัญหาการจับคู่เหมือนของสายอักขระได้อีกกรณีนอกจากการใช้สำหรับจำแนกข้อมูล ในปัญหาการจับคู่เหมือนนั้นสามารถประยุกต์ใช้ได้หลากหลายเช่น การหาสายอักขระคำเดิมท้าย การหาสายอักขระนำหน้า การหาสายอักขระเอกลักษณ์และสายอักขระเกิดซ้ำ การนับจำนวนสายอักขระ การหาสายอักขระเกิดซ้ำยาวที่สุด การหาสายอักขระเอกลักษณ์ยาวที่สุด อย่างไรก็ตามมีประเด็นที่น่าสนใจอย่างหนึ่งคือ ระหว่างแบบจำลองในงานวิจัยนี้มีความคล้ายคลึงหรือเทียบเท่ากันกับต้นไม้คำเดิมท้ายในมิติใดบ้าง สามารถจะพิสูจน์ด้วยทฤษฎีได้อย่างไร ก็เป็นงานวิจัยในเชิงทฤษฎีที่น่าสนใจอีกหัวข้อหนึ่ง

ประเด็นที่น่าสนใจสำหรับงานวิจัยต่อยอดอีกประการหนึ่งคือ งานวิจัยนี้ใช้การคำนวณหาความน่าจะเป็นของสายอักขระและเทคนิคการปรับเรียบอย่างง่าย โดยใช้สถานะรอทำการในแต่ละรอบเพื่อสำรองการหาความน่าจะเป็นที่ไม่เป็น 0 แต่เทคนิคการปรับเรียบยังมีวิธีอื่นที่น่าสนใจ ซึ่งอาจจะทำให้การคำนวณความน่าจะเป็นสำหรับการจำแนกข้อมูลมีความแม่นยำมากยิ่งขึ้น การทดลองใช้การปรับเรียบด้วยวิธีอื่นและข้อมูลอินพุตอื่นๆก็เป็นหัวข้อที่น่าสนใจ

เนื่องจากการเรียนรู้ด้วยแบบจำลองนี้ไม่จำกัดขนาดโครงสร้างในการเรียนรู้ นั่นหมายความว่าข้อมูลที่จะเรียนรู้เมื่อเข้ามาใหม่แล้ว อาจจะทำให้หน่วยความจำเพิ่มขึ้นได้ไม่จำกัดจำนวน ดังนั้นในการประยุกต์ใช้งานในบางกรณีอาจจะต้องมีการจำกัดโครงสร้างไว้เท่าที่กำหนดซึ่งจะมีวิธีการทำอย่างไรให้เกิดประสิทธิภาพสูงสุด

สำหรับการนำแบบจำลองนี้ไปประยุกต์ใช้กับโจทย์ปัญหาในการจำแนกข้อมูลลำดับเชิงสัญลักษณ์ที่น่าสนใจ เช่น การจำแนกกลุ่มข้อมูลดีเอ็นเอ การตรวจโรคจากคลื่นไฟฟ้าหัวใจ การตรวจจับการบุกรุกทางคอมพิวเตอร์ การรู้จำภาษาธรรมชาติ การพยากรณ์ภูมิอากาศ การรู้จำเสียง การรู้จำภาพ และอื่นๆอีกมากมายในทางปัญญาประดิษฐ์ ก็เป็นปัญหาที่ท้าทายและน่าสนใจ และน่าจะทำได้มีประสิทธิภาพดีเมื่อเทียบกับเครื่องมืออื่นๆที่มีอยู่ในปัจจุบันเนื่องจากแบบจำลองที่เสนอในงานวิจัยนี้สามารถตรวจความสัมพันธ์ของข้อมูลที่ขึ้นอยู่กับช่วงระยะยาวได้ตามความยาวของสายอักขระเอกลักษณ์และสายอักขระเกิดซ้ำที่พบ ดังนั้นจึงสามารถหาความสัมพันธ์ระยะยาวของข้อมูลต่างๆได้ดีพอสมควร

แบบจำลองไม่จำกัดเฉพาะสายอักขระของตัวอักษรแต่สามารถประยุกต์ใช้คำแทนตัวอักษรได้ จะได้เป็นสายอักขระของคำหรือประโยค ซึ่งจะทำให้ขยายขีดความสามารถของแบบจำลองให้กว้างขึ้น เช่น การใช้แบบจำลองกับภาษาธรรมชาติ ซึ่งเป็นการเรียงต่อกันของคำในรูปแบบของประโยค นอกจากนี้การใช้ฐานข้อมูลเพื่อจัดเก็บแบบจำลองเพื่อนำไปใช้ในระยะยาวแทนการจัดเก็บข้อมูลดิบก็เป็นเรื่องเชิงปฏิบัติที่น่าศึกษาหาข้อดีข้อเสียต่อไป ข้อเสียเปรียบของแบบจำลองในงานวิจัยนี้คือ จำนวนสถานะที่ค่อนข้างมากเมื่อจำนวนอินพุตมากขึ้น อย่างไรก็ตามจำนวนสถานะดีที่สุด $O(n)$ และไม่มากไปกว่า $O(n^2)$ และเนื่องจากเทคโนโลยีทางหน่วยความจำในปัจจุบันนั้นมีศักยภาพมากพอที่จะรองรับการเติบโตอันไม่จำกัดของข้อมูล ข้อเสียเปรียบของแบบจำลองนี้จึงน่าจะสามารถยอมรับได้ในเชิงปฏิบัติ อย่างไรก็ตามจำนวนสถานะที่เพิ่มขึ้นนั้นมีความสำคัญต่อคุณสมบัติของแบบจำลองสำหรับปรับความรู้เมื่อได้รับข้อมูลส่วนเพิ่ม เว้นแต่ว่าการเรียนรู้ของแบบจำลองนั้นจะไม่ต้องมีการเรียนรู้ส่วนเพิ่มแล้ว การยุบสถานะให้น้อยลงก็เป็นส่วนงานวิจัยที่น่าสนใจ

ข้อมูลทดลองในงานวิจัยนี้สามารถดาวน์โหลดได้ที่ http://archive.ics.uci.edu/ml/machine-learning-databases/molecular-biology/promoter-gene-sequences/

บรรณานุกรม

Ade, R. R. & Deshmukh, P.R. (2013). Methods for Incremental Learning : A Survey. *International Journal of Data Mining & Knowledge Management Process* , 119-125.

Cavnar, W. B. & Trenkle, J. M. (1994). N-Gram-Based Text Categorization. *In Proceedings of SDAIR-94, 3rd Annual Symposium on Document Analysis and Information Retrieval* , 161–175.

Dietterich, T. G. (2002). Machine Learning for Sequential Data: A Review. *Proceedings of the Joint IAPR International Workshop on Structural, Syntactic, and Statistical Pattern Recognition* , 15–30.

Dupont, P., Denis, F., & Esposito, Y. (2005). Links Between Probabilistic Automata and Hidden Markov Models: Probability Distributions, Learning Models and Induction Algorithms. *Pattern Recogn.*, 38 (9), 1349–1371.

Geppert, E. & Hammer, B. (2016). Incremental learning algorithms and applications. *ESANN* , 357-368

Gold, E. M. (1967). Language identification in the limit. *Information and Control*, 10 (5), 447 - 474.

Ilie, L. & Smyth, W. F. (2011). Minimum Unique Substrings and Maximum Repeats. *Fundam. Inf*, 110 (1-4), 183–195.

Kermorvant, C. & Dupont, P. (2002). Improved Smoothing for Probabilistic Suffix Trees Seen As Variable Order Markov Chains. *Proceedings of the 13th European Conference on Machine Learning* , 185–194.

Khreich, W., Granger, E., Miri, A., & Sabourin, R. (2012). A Survey of Techniques for Incremental Learning of HMM Parameters. *Information Sciences*, 197 , 105–130.

Leslie, C., Eskin, E., & Noble, W. S. (2002). The spectrum kernel: a string kernel for SVM protein classification. *PAC Symp Biocomput*, , 564-575.

Lin, J., Keogh, E., Lonardi, S., & Chiu, B. (2003). A Symbolic Representation of Time Series, with Implications for Streaming Algorithms. *Proceedings of the 8th ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery* , 2–11

Lodhi, H., Saunders, C., Shawe-Taylor, J., Cristianini, N., & Watkins, C. (2002). Text classification using string kernels. *Journal of Machine Learining Research*, 2 , 419-444.

Manber, U. & Myers, G. (1990). Suffix Arrays: A New Method for On-line String Searches. *Proceedings of the First Annual ACM-SIAM Symposium on Discrete Algorithms* , 319–327.

Mansour, E., Allam, A., Skiadopoulos, S., & Kalnis, P. (2011). ERA: Efficient Serial and Parallel Suffix Tree Construction for Very Long Strings. Proc. *VLDB Endow* , 5 (1), 49–60.

Michailidis, P. D. & Margaritis, K. G. (2002). On-line Approximate String Searching Algorithms: Survey and Experimental Results. *International Journal of Computer Mathematics*, 79 (8), 867-888.

Rabiner, L. R. (1989). A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77 (2), 257-286.

Ron, D., Singer, Y., & Tishby, N. (1996). The Power of Amnesia: Learning Probabilistic Automata with Variable Memory Length. *Machine learning* , 117–149.

Towell, G., Shavlik, J., & Noordewier, M. (1990). Refinement of Approximate Domain Theories by Knowledge-Based Neural Networks. *In Proceedings of the Eighth National Conference on Artificial Intelligence* , 861-866.

Ukkonen, E. (1985). Algorithms for Approximate String Matching. *Inf. Control*, 64 (1-3), 100–118.

Ukkonen, E. (1995). On-line construction of suffix trees. *Algorithmica*, 14 (3), 249–260.

Valiant, L. G. (1984). A Theory of the Learnable. *Commun. ACM*, 27 (11), 1134–1142.

Vidal, E., Thollard, F., de la Higuera, C., Casacuberta, F., & Carrasco, R. C. (2005). Probabilistic Finite-State Machines-Part I. *IEEE Trans. Pattern Anal. Mach. Intell.*, 27 (7), 1013–1025.

Vidal, E., Thollard, F., de la Higuera, C., Casacuberta, F., & Carrasco, R. C. (2005). Probabilistic Finite-State Machines-Part II. *IEEE Trans. Pattern Anal. Mach. Intell.*, 27 (7), 1026–1039.

Weiner, P. (1973). Linear Pattern Matching Algorithms. *Proceedings of the 14th Annual Symposium on Switching and Automata Theory* , 1–11.

Xing, Z., Pei, J., & Keogh, E. (2010). A Brief Survey on Sequence Classification. *SIGKDD Explor. Newsl.*, 12 (1), 40–48.

Xing, Z., Pei, J., Dong, G., & Yu, P. S. (2008). Mining Sequence Classifiers for Early Prediction. *Proceedings of the {SIAM} International Conference on Data Mining*, 644–655.

Zdonik, S. B. (1993). Incremental Database Systems: Databases from the Ground Up. *SIGMOD Rec.*, 22 (2), 408–412.

ประวัตินักวิจัย

ชื่อ : จิตรกร พูลโพธิ์ทอง

ตำแหน่ง : ผู้ช่วยศาสตราจารย์

สังกัดหน่วยงาน : สาขาวิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยรามคำแหง

Email : jittakornp@hotmail.com

การศึกษา :

- ประถมศึกษาและมัธยมศึกษาตอนต้น โรงเรียนวีรศิลป์ ต. ท่าม่วง อ. ท่าม่วง จ. กาญจนบุรี
- มัธยมศึกษาตอนปลาย โรงเรียนสารสิทธิ์พิทยาลัย อ. บ้านโป่ง จ. ราชบุรี
- วิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมคอมพิวเตอร์ มหาวิทยาลัยมหิดล ปีการศึกษา 2543
- วิศวกรรมศาสตรมหาบัณฑิต สาขาวิชาวิศวกรรมคอมพิวเตอร์ จุฬาลงกรณ์มหาวิทยาลัย ปีการศึกษา 2548

ผลงานทางวิชาการ :

จิตรกร พูลโพธิ์ทอง 2553 GRE107 การโปรแกรมคอมพิวเตอร์สำหรับวิศวกร พิมพ์ครั้งที่ 1 กรุงเทพฯ สำนักพิมพ์มหาวิทยาลัยรามคำแหง

จิตรกร พูลโพธิ์ทอง 2554 แบบจำลองการจัดตารางสอนในสถาบันอุดมศึกษาและการระบุเงื่อนไขบังคับอย่างเป็นแบบแผน วารสารวิจัยรามคำแหง ฉบับวิทยาศาสตร์และเทคโนโลยี 14(2): 28-42

จิตรกร พูลโพธิ์ทอง 2559 ทฤษฎีการคำนวณ พิมพ์ครั้งที่ 1 กรุงเทพฯ สำนักพิมพ์มหาวิทยาลัยรามคำแหง

จิตรกร พูลโพธิ์ทอง 2560 GRE1007 การโปรแกรมคอมพิวเตอร์สำหรับวิศวกร พิมพ์ครั้งที่ 2(ฉบับปรับปรุงใหม่) กรุงเทพฯ สำนักพิมพ์มหาวิทยาลัยรามคำแหง

Pullpothong, J. Surarerk A. 2005 " Online learning stochastic automata in linear time" In proceeding of 9th National Computer Science and Engineering Conference: NCSEC2005