การปรับปรุงประสิทธิภาพของระบบแนะนำโดยการประยุกต์ใช้การจำแนกประเภทข้อมูลแบบหลายคำตอบและการลดจำนวนมิติในชุดคำตอบ

กันทรากร จิตต์หาญ

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชาวิศวกรรมข้อมูลขนาดใหญ่
วิทยาลัยนวัตกรรมด้านเทคโนโลยีและวิศวกรรมศาสตร์
มหาวิทยาลัยธุรกิจบัณฑิตย์
พ.ศ. 2563

Extreme Multi-label Classification with Label Space Reduction for Recommendation System

Kantarakorn Jitharn

A Thesis Submitted in Partial Fulfillment of the Requirements

for the Degree of Master of Engineering

Department of Big Data Engineering,

College of Innovative Technology and Engineering,

Dhurakij Pundit University

2020

ใบรับรองงานวิทยานิพนธ์

วิทยาลัยนวัตกรรมด้านเทคโนโลยีและวิศวกรรมศาสตร์ มหาวิทยาลัยธุรกิจบัณฑิตย์

ปริญญา วิศวกรรมศาสตรมหาบัณฑิต

หัวข้อวิทยานิพนธ์ การปรับปรุงประสิทธิภาพของระบบแนะนำโดยการประยุกต์ใช้การจำแนกประเภทข้อมูลแบบหลายคำตอบและการลดจำนวนมิติในชุดคำตอบ

เสนอโดย กันทรากร จิตต์หาญ

สาขาวิชา วิศวกรรมข้อมูลขนาดใหญ่

อาจารย์ที่ปรึกษาวิทยานิพนธ์ ดร.เอกสิทธิ์ พัชรวงศ์ศักดา

ได้พิจารณาเห็นชอบโดยคณะกรรมการสอบวิทยานิพนธ์แล้ว

.....ประธานกรรมการ

(ดร.สรรพฤทธิ์ มฤคทัต)

.....กรรมการและอาจารย์ที่ปรึกษาหลัก

(ดร.เอกสิทธิ์ พัชรวงศ์ศักดา)

.....กรรมการ

(ดร.ธนภัทร ฆังคะจิตร)

.....กรรมการ

(ผู้ช่วยศาสตราจารย์ ดร.ดวงใจ จิตคงชื่น)

วิทยาลัยนวัตกรรมด้านเทคโนโลยีและวิศวกรรมศาสตร์รับรองแล้ว

.....

(ผู้ช่วยศาสตราจารย์ ดร.ณรงค์เดช กีรติพรานนท์)

คณบดีวิทยาลัยนวัตกรรมด้านเทคโนโลยีและวิศวกรรมศาสตร์

วันที่ ...27.... เดือน ...เมษายน... พ.ศ. ...2563.....

หัวข้อวิทยานิพนธ์ การปรับปรุงประสิทธิภาพของระบบแนะนำโดยการประยุกต์ใช้การจำแนกประเภทข้อมูลแบบหลายคำตอบและการลดจำนวนมิติในชุดคำตอบ
ชื่อผู้เขียน กันทรากร จิตต์หาญ
อาจารย์ที่ปรึกษา ดร.เอกสิทธิ์ พัชรวงศ์ศักดา
สาขาวิชา วิศวกรรมข้อมูลขนาดใหญ่
ปีการศึกษา 2562

บทคัดย่อ

ระบบการแนะนำ (recommendation system) เป็นระบบหนึ่งที่มีการใช้งานกันอย่างกว้างขวางในหลากหลายธุรกิจ เช่น การแนะนำหนังสือ การแนะนำภาพยนตร์ หรือ การแนะนำสถานที่ท่องเที่ยวต่าง ๆ แม้ว่าระบบนี้จะได้รับความนิยมแต่ก็ยังประสบปัญหาที่ท้าทายอีก 2 ประการ ได้แก่ ปัญหาการขาดข้อมูลตั้งต้น (cold start) และปัญหาการมีมิติจำนวนมาก (high-dimensionality) โดยปัญหาแรกนั้นเกิดจากการที่ระบบไม่สามารถแนะนำสินค้าหรือบริการที่เหมาะสมให้กับผู้ใช้งานใหม่ที่เข้ามาในระบบได้ เนื่องจากขาดข้อมูลประวัติการทำกิจกรรมต่าง ๆ ของผู้ใช้งานใหม่เหล่านั้น ส่วนปัญหาที่สองเกิดจากการที่สินค้าหรือบริการมีจำนวนหลากหลายมากจนยากที่จะแนะนำได้อย่างถูกต้อง ลักษณะของระบบการแนะนำนี้สามารถแปลงให้อยู่ในรูปแบบของการจำแนกประเภทข้อมูลแบบหลายคำตอบ (Multi-label Classification) ได้โดยให้สินค้าหรือบริการเป็นคำตอบ (Label) ต่าง ๆ

งานวิจัยนี้มีวัตถุประสงค์เพื่อแก้ปัญหาทั้งสองประการข้างต้นโดยการใช้เทคนิคการจำแนกประเภทข้อมูลแบบหลายคำตอบและเทคนิคการลดจำนวนมิติในส่วนของคำตอบ (Label Space) วิธีการที่นำเสนอเริ่มจากใช้เทคนิคซิงกูลาร์ แวลู ดีคอมโพสิชัน (Singular Value Decomposition) หรือ SVD ทำการลดจำนวนมิติในส่วนของคำตอบเสียก่อน หลังจากนั้นใช้เทคนิคการเรียนรู้เชิงลึก (Deep Learning) หาปัจจัย (Feature) ที่สำคัญจากข้อมูลและทำการจำแนกประเภทข้อมูลออกมาเป็นหลายคำตอบ หลังจากนั้นทำการแปลงจำนวนมิติในส่วนของคำตอบให้กลับไปเป็นจำนวนเท่าเดิมก่อนที่จะนำเสนอสินค้าหรือบริการให้กับผู้ใช้งานต่อไป

จากการทดสอบด้วยชุดข้อมูลทดสอบจากเว็บไซต์ สแทคโอเวอร์โฟลว์ (Stackoverflow) พบว่าวิธีการที่นำเสนอมีประสิทธิภาพดีขึ้นในแง่ของ รีคอลล์ที่จำนวน M ตัวแรก (Recall@M) และความน่าจะเป็นสะสมที่ลดทอนด้วยค่าปกติ (NDCG@M) เมื่อความหนาแน่น (Label Density) ของข้อมูลมากกว่า 0.0004 และใช้การลดจำนวนมิติลงเหลือ 50% และ 80% ของจำนวนตั้งต้น

Thesis Title	Extreme Multi-Label Classification with Label Space Reduction for Recommendation System
Author	Kantarakorn Jitharn
Thesis Advisor	Dr. Eakasit Pacharawongsakda
Department	Big Data Engineering
Academic Year	2019

ABSTRACT

The recommendation system has been widely used in various areas, e.g., entertainment, education, and travel. However, this technique faces two main challenges, which are Cold Start and High-Dimensionality problems. The Cold Start happens when the system doesn't have enough profile of the new users; therefore, the system cannot suggest products to them. The second issue comes from the fact that there are a lot of distinct users or products to be recommended. Recently, Extreme Multi-label Classification (XMLC) has been applied to the recommendation system and addressed the Cold Start issue. However, the previous method still has a High-Dimensionality issue. In this research, we proposed a new approach, namely XMLC-PAO, which integrating label space reduction with XMLC.

In more detail, we transformed the recommendation problem to XMLC and applied Singular Value Decomposition (SVD) to generate reducing operator of label space (in this case was products/users). For the feature space, Deep Learning technique has been used to extract features from the text. From the experiments with Stackoverflow online forums dataset, we found that the XMLC-PAO showed better performance in terms of RECALL@M and NDCG@M when label density is more than 0.0004 and the dimensions were reduced to 50% and 80% of the original size.

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงไปได้ด้วยการให้ความช่วยเหลือแนะนำของ ดร.เอกสิทธิ์ พัชรวงศ์ศักดา ซึ่งเป็นอาจารย์ที่ปรึกษาวิทยานิพนธ์ ที่ได้กรุณาที่ให้คำแนะนำข้อคิดเห็นตรวจสอบ และแก้ไขร่างวิทยานิพนธ์มาโดยตลอด ผู้เขียนจึงขอกราบขอบพระคุณไว้ ณ โอกาสนี้

ผู้เขียนขอกราบขอบพระคุณ ดร.สรรพฤทธิ์ มฤคทัต ที่กรุณาให้เกียรติเป็นประธานโดยมี ดร. ธนภัทร ฆังคะจิตร, ผู้ช่วยศาสตราจารย์ ดร. ดวงใจ จิตคงชื่น และ ดร.เอกสิทธิ์ พัชรวงศ์ศักดา เป็นคณะกรรมการในการสอบวิทยานิพนธ์ ซึ่งได้กรุณาให้คำแนะนำในการปรับปรุงวิทยานิพนธ์ให้สมบูรณ์ครบถ้วน ตลอดจน นางสาวกุลธิดา รอดบุญ รวมถึงเจ้าหน้าที่บัณฑิตมหาวิทยาลัยธุรกิจบัณฑิตย์ ทุกท่านที่ให้ความสะดวกด้านอำนวยการ และประสานงาน ในการทำวิทยานิพนธ์ให้ผู้เขียนตลอดมาตลอดจนค้นคว้าหาข้อมูลในการจัดทำวิทยานิพนธ์ของผู้เขียนครั้งนี้สำเร็จลุล่วงไปด้วยดี

นอกจากนี้ขอกราบขอบพระคุณ Kishaloy Halder ผู้เขียนบทความเรื่อง Cold Start Thread Recommendation as Extreme Multi-label Classification (Kishaloy Halder, 2018) ที่ยินยอมให้ใช้โปรแกรมภาษาไพทอน (Python) ซึ่งสร้างขึ้นสำหรับการเขียนบทความดังกล่าว เพื่อใช้เป็นโปรแกรมตั้งต้นในการพัฒนางานวิจัยนี้

ท้ายนี้ผู้เขียนขอน้อมรำลึกถึงอำนาจบารมีของคุณพระศรีรัตนตรัย และสิ่งศักดิ์สิทธิ์ทั้งหลายที่อยู่ในสากลโลก อันเป็นที่พึ่งให้ผู้เขียนมีสติปัญญาในการจัดทำวิทยานิพนธ์ให้สำเร็จลุล่วงไปด้วยดี ผู้เขียนขอให้เป็นกตเวทิตาแด่บิดา มารดา ครอบครัวของผู้เขียน ตลอดจนผู้เขียนหนังสือ และบทความต่าง ๆ ที่ให้ความรู้แก่ผู้เขียนจนสามารถทำให้วิทยานิพนธ์ฉบับนี้สำเร็จได้ด้วยดี

กันทรากร จิตต์หาญ

สารบัญ

หน้า

บทคัดย่อภาษาไทย…….…..... ฆ

บทคัดย่อภาษาอังกฤษ…….....…....……..... ง

กิตติกรรมประกาศ……..... จ

สารบัญตาราง...……..……...…..... ฉ

สารบัญภาพ ……..……..... ฌ

บทที่

1. บทนำ…….……..……..…..... 1

1.1 ที่มาและความสำคัญของปัญหา……..……..……..…..... 2

1.2 วัตถุประสงค์……..……..……...…….…..... 2

1.3 ขอบเขตงานวิจัย…….……..……...……..... 2

1.4 สมมติฐานของงานวิจัย…..……..…..... 3

1.5 ประโยชน์ที่คาดว่าจะได้รับ……..…..... 3

1.6 นิยามศัพท์……...……..……...…..... 3

2. แนวคิด ทฤษฎี และผลงานวิจัยที่เกี่ยวข้อง……..……..…….. 5

2.1 ระบบการแนะนำ……..…. 5

2.2 ปัญหาการขาดข้อมูลตั้งต้น ……..…… 9

2.3 การจำแนกประเภทข้อมูลหลายคำตอบ...…… 10

2.4 การแทนค่าคำ …….. 11

2.5 โครงข่ายประสาทแบบวนรอบ …… 12

2.6 เทคนิคซิงกูลาร์ แวลู ดีคอมโพสิชัน…….. 14

2.7 งานวิจัยที่เกี่ยวข้อง…… 15

3. ระเบียบวิธีวิจัย ……... 17

3.1 ลักษณะของชุดข้อมูล…….. 17

3.2 การเตรียมข้อมูล ……. 19

3.3 การคำนวณตัวดำเนินการแปลงมิติเชิงเส้น…… 23

3.4 โครงสร้างของแบบจำลอง…… 24

สารบัญ (ต่อ)

หน้า

3.5 การวัดประสิทธิภาพของแบบจำลอง…… 30
3.6 ปริมาณทางสถิติสำหรับระบุสถานะของชุดข้อมูลคำตอบในการจำแนกประเภทข้อมูลหลายคำตอบ……...……...…… 33
3.7 แนวทางการวิจัย……...……...……. 34
4. ผลการศึกษา....…….... 40
4.1 ตรวจสอบสมมติฐาน โดยการสุ่มเลือกข้อมูล 5000 ตัว ทั้งการเลือกข้อมูลแบบเชิงเส้น และการเลือกข้อมูลแบบสุ่ม……..... 40
4.2 ตรวจสอบสมมติฐาน โดยการเลือกข้อมูลแบบเชิงเส้น 6000 ตัว……..... 44
4.3 ตรวจสอบสมมติฐาน โดยการเลือกข้อมูล 7000 ตัวแรก……..... 46
4.4 ตรวจสอบสมมติฐาน โดยใช้ข้อมูลทั้งหมด……..... 46
4.5 เปรียบเทียบผลชุดข้อมูลทดสอบของแบบจำลองที่ดีที่สุด สำหรับแต่ละกลุ่มตัวอย่าง……..... 47
ตารางผลการทดลองสำหรับข้อมูลชุดทดสอบ……..... 54
ตารางผลการทดลองสำหรับข้อมูลชุดเรียนรู้……..... 91
5. บทสรุปและข้อเสนอแนะ……..... 128
5.1 อภิปรายผลการศึกษา……... 128
5.2 สรุปผลการศึกษา……..... 130
5.3 ข้อเสนอแนะ……..... 131
บรรณานุกรม……...……. 132
ภาคผนวก……...……..... 135
ก โปรแกรมภาษาไพทอนและจำนวนพารามิเตอร์สำหรับแบบจำลองของแต่ละชุดข้อมูล..... 136
ข ผลงานตีพิมพ์..... 161
ประวัติผู้เขียน..... 169

สารบัญตาราง

ตารางที่ **หน้า**

2.1 ตารางความสัมพันธ์ (Utility Matrix)..... 6

2.1 ความน่าจะเป็นของคำที่จะเกิดร่วมกัน สำหรับคำว่า “ice” และ “stream” ค่าความน่าจะเป็นที่มากที่สุดจะสัมพันธ์กับคุณสมบัติของ “ice” ส่วนค่าความน่าจะเป็นที่น้อยที่สุดจะสัมพันธ์กับคุณสมบัติของ “stream”..... 12

3.1 แสดงรายละเอียด และคุณสมบัติของแต่ละชุดข้อมูลคำตอบ..... 35

3.2 แสดงจำนวนคำเฉลี่ยในกระทู้ สำหรับชุดเรียนรู้ และชุดทดสอบ ของแต่ละชุดข้อมูล.....…….... 36

4.1. เปรียบเทียบผลข้อมูล Linear 1-5000 ชุดทดสอบ สำหรับสัดส่วนการลดมิติ 0.5 ระหว่างแบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับ แบบจำลองตั้งต้น..... 55

4.2 เปรียบเทียบผลข้อมูล Linear 1-5000 ชุดทดสอบ สำหรับสัดส่วนการลดมิติ 0.8 ระหว่างแบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับ แบบจำลองตั้งต้น..... 55

4.3 เปรียบเทียบผลข้อมูล Linear 1-5000 ชุดทดสอบ สำหรับสัดส่วนการลดมิติ 1.0 ระหว่างแบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับ แบบจำลองตั้งต้น..... 56

4.4 เปรียบเทียบผลข้อมูล Linear 201-5200 ชุดทดสอบ สำหรับสัดส่วนการลดมิติ 0.5 ระหว่างแบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับ แบบจำลองตั้งต้น..... 57

4.5 เปรียบเทียบผลข้อมูล Linear 201-5200 ชุดทดสอบ สำหรับสัดส่วนการลดมิติ 0.8 ระหว่างแบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับ แบบจำลองตั้งต้น..... 57

4.6 เปรียบเทียบผลข้อมูล Linear 201-5200 ชุดทดสอบ สำหรับสัดส่วนการลดมิติ 1.0 ระหว่างแบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับ แบบจำลองตั้งต้น..... 58

4.7 เปรียบเทียบผลข้อมูล Linear 801-5800 ชุดทดสอบ สำหรับสัดส่วนการลดมิติ 0.5 ระหว่างแบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับ แบบจำลองตั้งต้น..... 59

4.8 เปรียบเทียบผลข้อมูล Linear 801-5800 ชุดทดสอบ สำหรับสัดส่วนการลดมิติ 0.8 ระหว่างแบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับ แบบจำลองตั้งต้น..... 59

4.9 เปรียบเทียบผลข้อมูล Linear 801-5800 ชุดทดสอบ สำหรับสัดส่วนการลดมิติ 1.0 ระหว่างแบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับ แบบจำลองตั้งต้น..... 60

สารบัญตาราง (ต่อ)

ตารางที่ **หน้า**

4.10 เปรียบเทียบผลข้อมูล Linear 5001-10000 ชุดทดสอบ สำหรับสัดส่วนการลดมิติ 0.5 ระหว่างแบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับ แบบจำลองตั้งต้น..... 61
4.11 เปรียบเทียบผลข้อมูล Linear 5001-10000 ชุดทดสอบ สำหรับสัดส่วนการลดมิติ 0.8 ระหว่างแบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับ แบบจำลองตั้งต้น..... 61
4.12 เปรียบเทียบผลข้อมูล Linear 5001-10000 ชุดทดสอบ สำหรับสัดส่วนการลดมิติ 1.0 ระหว่างแบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับ แบบจำลองตั้งต้น..... 62
4.13 เปรียบเทียบผลข้อมูล Linear 10001-15000 ชุดทดสอบ สำหรับสัดส่วนการลดมิติ 0.5 ระหว่างแบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับ แบบจำลองตั้งต้น..... 63
4.14 เปรียบเทียบผลข้อมูล Linear 10001-15000 ชุดทดสอบ สำหรับสัดส่วนการลดมิติ 0.8 ระหว่างแบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับ แบบจำลองตั้งต้น..... 63
4.15 เปรียบเทียบผลข้อมูล Linear 10001-15000 ชุดทดสอบ สำหรับสัดส่วนการลดมิติ 1.0 ระหว่างแบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับ แบบจำลองตั้งต้น..... 64
4.16 เปรียบเทียบผลข้อมูล Linear 12201-17200 ชุดทดสอบ สำหรับสัดส่วนการลดมิติ 0.5 ระหว่างแบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับ แบบจำลองตั้งต้น..... 65
4.17 เปรียบเทียบผลข้อมูล Linear 12201-17200 ชุดทดสอบ สำหรับสัดส่วนการลดมิติ 0.8 ระหว่างแบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับ แบบจำลองตั้งต้น..... 65
4.18 เปรียบเทียบผลข้อมูล Linear 12201-17200 ชุดทดสอบ สำหรับสัดส่วนการลดมิติ 1.0 ระหว่างแบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับ แบบจำลองตั้งต้น..... 66
4.19 เปรียบเทียบผลข้อมูล Linear 15001-20000 ชุดทดสอบ สำหรับสัดส่วนการลดมิติ 0.5 ระหว่างแบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับ แบบจำลองตั้งต้น..... 67
4.20 เปรียบเทียบผลข้อมูล Linear 15001-20000 ชุดทดสอบ สำหรับสัดส่วนการลดมิติ 0.8 ระหว่างแบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับ แบบจำลองตั้งต้น..... 67
4.21 เปรียบเทียบผลข้อมูล Linear 15001-20000 ชุดทดสอบ สำหรับสัดส่วนการลดมิติ 1.0 ระหว่างแบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับ แบบจำลองตั้งต้น..... 68
4.22 เปรียบเทียบผลข้อมูล Linear 20001-25000 ชุดทดสอบ สำหรับสัดส่วนการลดมิติ 0.5 ระหว่างแบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับ แบบจำลองตั้งต้น..... 69

สารบัญตาราง (ต่อ)

ตารางที่ **หน้า**

4.23 เปรียบเทียบผลข้อมูล Linear 20001-25000 ชุดทดสอบ สำหรับสัดส่วนการลดมิติ 0.8 ระหว่างแบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับ แบบจำลองตั้งต้น..... 69
4.24 เปรียบเทียบผลข้อมูล Linear 20001-25000 ชุดทดสอบ สำหรับสัดส่วนการลดมิติ 1.0 ระหว่างแบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับ แบบจำลองตั้งต้น..... 70
4.25 เปรียบเทียบผลข้อมูล Linear 25001-27932 (2933 ตัวอย่าง) ชุดทดสอบ สำหรับสัดส่วนการลดมิติ 0.5 ระหว่างแบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับ แบบจำลองตั้งต้น..... 71
4.26 เปรียบเทียบผลข้อมูล Linear 25001-27932 (2933 ตัวอย่าง) ชุดทดสอบ สำหรับสัดส่วนการลดมิติ 0.8 ระหว่างแบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับ แบบจำลองตั้งต้น..... 71
4.27 เปรียบเทียบผลข้อมูล Linear 25001-27932 (2933 ตัวอย่าง) ชุดทดสอบ สำหรับสัดส่วนการลดมิติ 1.0 ระหว่างแบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับ แบบจำลองตั้งต้น..... 72
4.28 เปรียบเทียบผลข้อมูล Random seed-42 ชุดทดสอบ สำหรับสัดส่วนการลดมิติ 0.5 ระหว่างแบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับ แบบจำลองตั้งต้น..... 73
4.29 เปรียบเทียบผลข้อมูล Random seed-42 ชุดทดสอบ สำหรับสัดส่วนการลดมิติ 0.8 ระหว่างแบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับ แบบจำลองตั้งต้น..... 73
4.30 เปรียบเทียบผลข้อมูล Random seed-42 ชุดทดสอบ สำหรับสัดส่วนการลดมิติ 1.0 ระหว่างแบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับ แบบจำลองตั้งต้น..... 74
4.31 เปรียบเทียบผลข้อมูล Random seed-101 ชุดทดสอบ สำหรับสัดส่วนการลดมิติ 0.5 ระหว่างแบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับ แบบจำลองตั้งต้น..... 75
4.32 เปรียบเทียบผลข้อมูล Random seed-101 ชุดทดสอบ สำหรับสัดส่วนการลดมิติ 0.8 ระหว่างแบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับ แบบจำลองตั้งต้น..... 75
4.33 เปรียบเทียบผลข้อมูล Random seed-101 ชุดทดสอบ สำหรับสัดส่วนการลดมิติ 1.0 ระหว่างแบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับ แบบจำลองตั้งต้น..... 76

สารบัญตาราง (ต่อ)

ตารางที่ **หน้า**

4.34 เปรียบเทียบผลข้อมูล Linear 1-6000 ชุดทดสอบ สำหรับสัดส่วนการลดมิติ 0.5 ระหว่างแบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับ แบบจำลองตั้งต้น..... 77
4.35 เปรียบเทียบผลข้อมูล Linear 1-6000 ชุดทดสอบ สำหรับสัดส่วนการลดมิติ 0.8 ระหว่างแบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับ แบบจำลองตั้งต้น..... 77
4.36 เปรียบเทียบผลข้อมูล Linear 1-6000 ชุดทดสอบ สำหรับสัดส่วนการลดมิติ 1.0 ระหว่างแบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับ แบบจำลองตั้งต้น..... 78
4.37 เปรียบเทียบผลข้อมูล Linear 6001-12000 ชุดทดสอบ สำหรับสัดส่วนการลดมิติ 0.5 ระหว่างแบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับ แบบจำลองตั้งต้น..... 79
4.38 เปรียบเทียบผลข้อมูล Linear 6001-12000 ชุดทดสอบ สำหรับสัดส่วนการลดมิติ 0.8 ระหว่างแบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับ แบบจำลองตั้งต้น..... 79
4.39 เปรียบเทียบผลข้อมูล Linear 6001-12000 ชุดทดสอบ สำหรับสัดส่วนการลดมิติ 1.0 ระหว่างแบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับ แบบจำลองตั้งต้น..... 80
4.40 เปรียบเทียบผลข้อมูล Linear 12001-18000 ชุดทดสอบ สำหรับสัดส่วนการลดมิติ 0.5 ระหว่างแบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับ แบบจำลองตั้งต้น..... 81
4.41 เปรียบเทียบผลข้อมูล Linear 12001-18000 ชุดทดสอบ สำหรับสัดส่วนการลดมิติ 0.8 ระหว่างแบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับ แบบจำลองตั้งต้น..... 81
4.42 เปรียบเทียบผลข้อมูล Linear 12001-18000 ชุดทดสอบ สำหรับสัดส่วนการลดมิติ 1.0 ระหว่างแบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับ แบบจำลองตั้งต้น..... 82
4.43 เปรียบเทียบผลข้อมูล Linear 18001-24000 ชุดทดสอบ สำหรับสัดส่วนการลดมิติ 0.5 ระหว่างแบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับ แบบจำลองตั้งต้น..... 83
4.44 เปรียบเทียบผลข้อมูล Linear 18001-24000 ชุดทดสอบ สำหรับสัดส่วนการลดมิติ 0.8 ระหว่างแบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับ แบบจำลองตั้งต้น..... 83
4.45 เปรียบเทียบผลข้อมูล Linear 18001-24000 ชุดทดสอบ สำหรับสัดส่วนการลดมิติ 1.0 ระหว่างแบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับ แบบจำลองตั้งต้น..... 84

สารบัญตาราง (ต่อ)

ตารางที่ **หน้า**

4.46 เปรียบเทียบผลข้อมูล Linear 24001-27932 (3933 ตัวอย่าง) ชุดทดสอบ สำหรับสัดส่วนการลดมิติ 0.5 ระหว่างแบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับ แบบจำลองตั้งต้น..... 85
4.47 เปรียบเทียบผลข้อมูล Linear 24001-27932 (3933 ตัวอย่าง) ชุดทดสอบ สำหรับสัดส่วนการลดมิติ 0.8 ระหว่างแบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับ แบบจำลองตั้งต้น..... 85
4.48 เปรียบเทียบผลข้อมูล Linear 24001-27932 (3933 ตัวอย่าง) ชุดทดสอบ สำหรับสัดส่วนการลดมิติ 1.0 ระหว่างแบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับ แบบจำลองตั้งต้น..... 86
4.49 เปรียบเทียบผลข้อมูล Linear 1-7000 ชุดทดสอบ สำหรับสัดส่วนการลดมิติ 0.5 ระหว่างแบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับ แบบจำลองตั้งต้น..... 87
4.50 เปรียบเทียบผลข้อมูล Linear 1-7000 ชุดทดสอบ สำหรับสัดส่วนการลดมิติ 0.8 ระหว่างแบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับ แบบจำลองตั้งต้น..... 87
4.51 เปรียบเทียบผลข้อมูล Linear 1-7000 ชุดทดสอบ สำหรับสัดส่วนการลดมิติ 1.0 ระหว่างแบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับ แบบจำลองตั้งต้น..... 88
4.52 เปรียบเทียบผลข้อมูล Linear 1-27932 ชุดทดสอบ สำหรับสัดส่วนการลดมิติ 0.5 ระหว่างแบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับ แบบจำลองตั้งต้น..... 89
4.53 เปรียบเทียบผลข้อมูล Linear 1-27932 ชุดทดสอบ สำหรับสัดส่วนการลดมิติ 0.8 ระหว่างแบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับ แบบจำลองตั้งต้น..... 89
4.54 เปรียบเทียบผลข้อมูล Linear 1-27932 ชุดทดสอบ สำหรับสัดส่วนการลดมิติ 1.0 ระหว่างแบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับ แบบจำลองตั้งต้น..... 90
4.55 เปรียบเทียบผลข้อมูล Linear 1-5000 ชุดเรียนรู้ สำหรับสัดส่วนการลดมิติ 0.5 ระหว่างแบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับ แบบจำลองตั้งต้น..... 92
4.56 เปรียบเทียบผลข้อมูล Linear 1-5000 ชุดเรียนรู้ สำหรับสัดส่วนการลดมิติ 0.8 ระหว่างแบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับ แบบจำลองตั้งต้น..... 92

สารบัญตาราง (ต่อ)

ตารางที่ **หน้า**

4.57 เปรียบเทียบผลข้อมูล Linear 1-5000 ชุดเรียนรู้ สำหรับสัดส่วนการลดมิติ 1.0 ระหว่างแบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับ แบบจำลองตั้งต้น..... 93
4.58 เปรียบเทียบผลข้อมูล Linear 201-5200 ชุดเรียนรู้ สำหรับสัดส่วนการลดมิติ 0.5 ระหว่างแบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับ แบบจำลองตั้งต้น..... 94
4.59 เปรียบเทียบผลข้อมูล Linear 201-5200 ชุดเรียนรู้ สำหรับสัดส่วนการลดมิติ 0.8 ระหว่างแบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับ แบบจำลองตั้งต้น..... 94
4.60 เปรียบเทียบผลข้อมูล Linear 201-5200 ชุดเรียนรู้ สำหรับสัดส่วนการลดมิติ 1.0 ระหว่างแบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับ แบบจำลองตั้งต้น..... 95
4.61 เปรียบเทียบผลข้อมูล Linear 801-5800 ชุดเรียนรู้ สำหรับสัดส่วนการลดมิติ 0.5 ระหว่างแบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับ แบบจำลองตั้งต้น..... 96
4.62 เปรียบเทียบผลข้อมูล Linear 801-5800 ชุดเรียนรู้ สำหรับสัดส่วนการลดมิติ 0.8 ระหว่างแบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับ แบบจำลองตั้งต้น..... 96
4.63 เปรียบเทียบผลข้อมูล Linear 801-5800 ชุดเรียนรู้ สำหรับสัดส่วนการลดมิติ 1.0 ระหว่างแบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับ แบบจำลองตั้งต้น..... 97
4.64 เปรียบเทียบผลข้อมูล Linear 5001-10000 ชุดเรียนรู้ สำหรับสัดส่วนการลดมิติ 0.5 ระหว่างแบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับ แบบจำลองตั้งต้น..... 98
4.65 เปรียบเทียบผลข้อมูล Linear 5001-10000 ชุดเรียนรู้ สำหรับสัดส่วนการลดมิติ 0.8 ระหว่างแบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับ แบบจำลองตั้งต้น..... 98
4.66 เปรียบเทียบผลข้อมูล Linear 5001-10000 ชุดเรียนรู้ สำหรับสัดส่วนการลดมิติ 1.0 ระหว่างแบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับ แบบจำลองตั้งต้น..... 99
4.67 เปรียบเทียบผลข้อมูล Linear 10001-15000 ชุดเรียนรู้ สำหรับสัดส่วนการลดมิติ 0.5 ระหว่างแบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับ แบบจำลองตั้งต้น..... 100
4.68 เปรียบเทียบผลข้อมูล Linear 10001-15000 ชุดเรียนรู้ สำหรับสัดส่วนการลดมิติ 0.8 ระหว่างแบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับ แบบจำลองตั้งต้น..... 100
4.69 เปรียบเทียบผลข้อมูล Linear 10001-15000 ชุดเรียนรู้ สำหรับสัดส่วนการลดมิติ 1.0 ระหว่างแบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับ แบบจำลองตั้งต้น..... 101

สารบัญตาราง (ต่อ)

ตารางที่ **หน้า**

4.70 เปรียบเทียบผลข้อมูล Linear 12201-17200 ชุดเรียนรู้ สำหรับสัดส่วนการลดมิติ 0.5 ระหว่างแบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับ แบบจำลองตั้งต้น..... 102
4.71 เปรียบเทียบผลข้อมูล Linear 12201-17200 ชุดเรียนรู้ สำหรับสัดส่วนการลดมิติ 0.8 ระหว่างแบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับ แบบจำลองตั้งต้น..... 102
4.72 เปรียบเทียบผลข้อมูล Linear 12201-17200 ชุดเรียนรู้ สำหรับสัดส่วนการลดมิติ 1.0 ระหว่างแบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับ แบบจำลองตั้งต้น..... 103
4.73 เปรียบเทียบผลข้อมูล Linear 15001-20000 ชุดเรียนรู้ สำหรับสัดส่วนการลดมิติ 0.5 ระหว่างแบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับ แบบจำลองตั้งต้น..... 104
4.74 เปรียบเทียบผลข้อมูล Linear 15001-20000 ชุดเรียนรู้ สำหรับสัดส่วนการลดมิติ 0.8 ระหว่างแบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับ แบบจำลองตั้งต้น..... 104
4.75 เปรียบเทียบผลข้อมูล Linear 15001-20000 ชุดเรียนรู้ สำหรับสัดส่วนการลดมิติ 1.0 ระหว่างแบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับ แบบจำลองตั้งต้น..... 105
4.76 เปรียบเทียบผลข้อมูล Linear 20001-25000 ชุดเรียนรู้ สำหรับสัดส่วนการลดมิติ 0.5 ระหว่างแบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับ แบบจำลองตั้งต้น..... 106
4.77 เปรียบเทียบผลข้อมูล Linear 20001-25000 ชุดเรียนรู้ สำหรับสัดส่วนการลดมิติ 0.8 ระหว่างแบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับ แบบจำลองตั้งต้น..... 106
4.78 เปรียบเทียบผลข้อมูล Linear 20001-25000 ชุดเรียนรู้ สำหรับสัดส่วนการลดมิติ 1.0 ระหว่างแบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับ แบบจำลองตั้งต้น..... 107
4.79 เปรียบเทียบผลข้อมูล Linear 25001-27932 (2933 ตัวอย่าง) ชุดเรียนรู้ สำหรับสัดส่วนการลดมิติ 0.5 ระหว่างแบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับ แบบจำลองตั้งต้น..... 108
4.80 เปรียบเทียบผลข้อมูล Linear 25001-27932 (2933 ตัวอย่าง) ชุดเรียนรู้ สำหรับสัดส่วนการลดมิติ 0.8 ระหว่างแบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับ แบบจำลองตั้งต้น..... 108

สารบัญตาราง (ต่อ)

ตารางที่ **หน้า**

4.81 เปรียบเทียบผลข้อมูล Linear 25001-27932 (2933 ตัวอย่าง) ชุดเรียนรู้ สำหรับสัดส่วนการลดมิติ 1.0 ระหว่างแบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับ แบบจำลองตั้งต้น..... 109
4.82 เปรียบเทียบผลข้อมูล Random seed-42 ชุดเรียนรู้ สำหรับสัดส่วนการลดมิติ 0.5 ระหว่างแบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับ แบบจำลองตั้งต้น..... 110
4.83 เปรียบเทียบผลข้อมูล Random seed-42 ชุดเรียนรู้ สำหรับสัดส่วนการลดมิติ 0.8 ระหว่างแบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับ แบบจำลองตั้งต้น..... 110
4.84 เปรียบเทียบผลข้อมูล Random seed-42 ชุดเรียนรู้ สำหรับสัดส่วนการลดมิติ 1.0 ระหว่างแบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับ แบบจำลองตั้งต้น..... 111
4.85 เปรียบเทียบผลข้อมูล Random seed-101 ชุดเรียนรู้ สำหรับสัดส่วนการลดมิติ 0.5 ระหว่างแบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับ แบบจำลองตั้งต้น..... 112
4.86 เปรียบเทียบผลข้อมูล Random seed-101 ชุดเรียนรู้ สำหรับสัดส่วนการลดมิติ 0.8 ระหว่างแบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับ แบบจำลองตั้งต้น..... 112
4.87 เปรียบเทียบผลข้อมูล Random seed-101 ชุดเรียนรู้ สำหรับสัดส่วนการลดมิติ 1.0 ระหว่างแบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับ แบบจำลองตั้งต้น..... 113
4.88 เปรียบเทียบผลข้อมูล Linear 1-6000 ชุดเรียนรู้ สำหรับสัดส่วนการลดมิติ 0.5 ระหว่างแบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับ แบบจำลองตั้งต้น..... 114
4.89 เปรียบเทียบผลข้อมูล Linear 1-6000 ชุดเรียนรู้ สำหรับสัดส่วนการลดมิติ 0.8 ระหว่างแบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับ แบบจำลองตั้งต้น..... 114
4.90 เปรียบเทียบผลข้อมูล Linear 1-6000 ชุดเรียนรู้ สำหรับสัดส่วนการลดมิติ 1.0 ระหว่างแบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับ แบบจำลองตั้งต้น..... 115
4.91 เปรียบเทียบผลข้อมูล Linear 6001-12000 ชุดเรียนรู้ สำหรับสัดส่วนการลดมิติ 0.5 ระหว่างแบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับ แบบจำลองตั้งต้น..... 116
4.92 เปรียบเทียบผลข้อมูล Linear 6001-12000 ชุดเรียนรู้ สำหรับสัดส่วนการลดมิติ 0.8 ระหว่างแบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับ แบบจำลองตั้งต้น..... 116

สารบัญตาราง (ต่อ)

ตารางที่ **หน้า**

4.93 เปรียบเทียบผลข้อมูล Linear 6001-12000 ชุดเรียนรู้ สำหรับสัดส่วนการลดมิติ 1.0 ระหว่างแบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับ แบบจำลองตั้งต้น..... 117
4.94 เปรียบเทียบผลข้อมูล Linear 12001-18000 ชุดเรียนรู้ สำหรับสัดส่วนการลดมิติ 0.5 ระหว่างแบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับ แบบจำลองตั้งต้น..... 118
4.95 เปรียบเทียบผลข้อมูล Linear 12001-18000 ชุดเรียนรู้ สำหรับสัดส่วนการลดมิติ 0.8 ระหว่างแบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับ แบบจำลองตั้งต้น..... 118
4.96 เปรียบเทียบผลข้อมูล Linear 12001-18000 ชุดเรียนรู้ สำหรับสัดส่วนการลดมิติ 1.0 ระหว่างแบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับ แบบจำลองตั้งต้น..... 119
4.97 เปรียบเทียบผลข้อมูล Linear 18001-24000 ชุดเรียนรู้ สำหรับสัดส่วนการลดมิติ 0.5 ระหว่างแบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับ แบบจำลองตั้งต้น..... 120
4.98 เปรียบเทียบผลข้อมูล Linear 18001-24000 ชุดเรียนรู้ สำหรับสัดส่วนการลดมิติ 0.8 ระหว่างแบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับ แบบจำลองตั้งต้น..... 120
4.99 เปรียบเทียบผลข้อมูล Linear 18001-24000 ชุดเรียนรู้ สำหรับสัดส่วนการลดมิติ 1.0 ระหว่างแบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับ แบบจำลองตั้งต้น..... 121
4.100 เปรียบเทียบผลข้อมูล Linear 24001-27932 (3933 ตัวอย่าง) ชุดเรียนรู้ สำหรับสัดส่วนการลดมิติ 0.5 ระหว่างแบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับ แบบจำลองตั้งต้น..... 122
4.101 เปรียบเทียบผลข้อมูล Linear 24001-27932 (3933 ตัวอย่าง) ชุดเรียนรู้ สำหรับสัดส่วนการลดมิติ 0.8 ระหว่างแบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับ แบบจำลองตั้งต้น..... 122
4.102 เปรียบเทียบผลข้อมูล Linear 24001-27932 (3933 ตัวอย่าง) ชุดเรียนรู้ สำหรับสัดส่วนการลดมิติ 1.0 ระหว่างแบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับ แบบจำลองตั้งต้น..... 123
4.103 เปรียบเทียบผลข้อมูล Linear 1-7000 ชุดเรียนรู้ สำหรับสัดส่วนการลดมิติ 0.5 ระหว่างแบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับ แบบจำลองตั้งต้น..... 124

สารบัญตาราง (ต่อ)

ตารางที่ **หน้า**

4.104 เปรียบเทียบผลข้อมูล Linear 1-7000 ชุดเรียนรู้ สำหรับสัดส่วนการลดมิติ 0.8 ระหว่างแบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับ แบบจำลองตั้งต้น..... 124

4.105 เปรียบเทียบผลข้อมูล Linear 1-7000 ชุดเรียนรู้ สำหรับสัดส่วนการลดมิติ 1.0 ระหว่างแบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับ แบบจำลองตั้งต้น..... 125

4.106 เปรียบเทียบผลข้อมูล Linear 1-27932 ชุดเรียนรู้ สำหรับสัดส่วนการลดมิติ 0.5 ระหว่างแบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับ แบบจำลองตั้งต้น..... 126

4.107 เปรียบเทียบผลข้อมูล Linear 1-27932 ชุดเรียนรู้ สำหรับสัดส่วนการลดมิติ 0.8 ระหว่างแบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับ แบบจำลองตั้งต้น..... 126

4.108 เปรียบเทียบผลข้อมูล Linear 1-27932 ชุดเรียนรู้ สัดส่วนการลดมิติ 1.0 ระหว่าง แบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับ แบบจำลองตั้งต้น..... 127

สารบัญรูปภาพ

ภาพที่ **หน้า**

2.1 แผนผังแสดงแนวคิดระบบการแนะนำแบบไฮบริดถ่วงน้ำหนัก..... 8
2.2 แผนผังแสดงแนวคิดระบบการแนะนำแบบไฮบริดตามลำดับขั้น..... 9
2.3 (ก) แสดงความสัมพันธ์ระหว่างผู้ใช้ (Users) และผลิตภัณฑ์ (Items) แบบกราฟสองฝั่ง (Bipartite Graph) (ข) ตารางความสัมพันธ์ระหว่างผู้ใช้แต่ละคน และผลิตภัณฑ์แต่ละอย่าง..... 10
2.4 แผนผังวิธีทางโครงข่ายประสาท แบบซีโบว์ และ สคิปแกรม..... 11
2.5 แผนผังโครงข่ายประสาทแบบวนรอบ..... 13
2.6 แผนผังโครงข่ายประสาทแบบวนรอบที่มีหน่วยความจำสั้นยาว..... 13
2.7 แผนผังโครงข่ายประสาทแบบวนรอบที่มีหน่วยความจำแบบย่อ..... 14
2.8 เทคนิคซิงกูลาร์ แวลู ดีคอมโพสิชัน สำหรับเมทริกซ์ A ที่มีขนาด $n \times d$..... 15
3.1 ตัวอย่างของชุดข้อมูลคำถาม..... 18
3.2 ตัวอย่างข้อความในคอลัมน์ Body ของชุดข้อมูลคำถาม..... 18
3.3 ตัวอย่างของชุดข้อมูลคำตอบ..... 18
3.4 ตัวอย่างข้อความในคอลัมน์ Body ของชุดข้อมูลคำตอบ..... 19
3.5 ตัวอย่างของชุดข้อมูลแท็ก..... 19
3.6 ตัวอย่างการเขียนฟังก์ชันสำหรับใช้ตัดส่วนที่ไม่จำเป็นของข้อความในคอลัมน์ Body ของชุดข้อมูลคำถาม..... 20
3.7 ข้อมูลขาเข้า และ ข้อมูลคำตอบ สำหรับการเรียนรู้ของแบบจำลอง..... 22
3.8 แผนผังแสดงหลักการมุมมองแบบไฮเปอร์คิวบ์ (hypercube view)..... 23
3.9 โครงสร้างของแบบจำลองตั้งต้น (Original Model) ในงานวิจัย Cold Start Thread Recommendation as Extreme Multi-label Classification (Kishaloy Halder, 2018) ซึ่งมีชุดคำตอบเป็น Y..... 26
3.10 แผนผังโครงสร้างแบบจำลองงานวิจัยด้วยวิธีการ “XMLC-PAO”..... 29
3.11 ความสัมพันธ์ของการแปลงมิติเชิงเส้นระหว่าง Y และ H..... 30

สารบัญรูปภาพ (ต่อ)

ภาพที่ **หน้า**

3.12 แผนผังแสดงหลักการคำนวณค่าส่วนกลับของลำดับ ของกระทู้ที่ t ในชุดข้อมูล ซึ่งมีค่าลำดับ r_t = 4 และ ส่วนกลับของลำดับ $1/r_t$ = 1/4..... 31

3.13 แผนผังแสดงหลักการคำนวณค่ารีคอลล์ที่จำนวน M ตัวแรก (Recall@M) เมื่อ M = 10 ของกระทู้ที่ t..... 32

4.1 แสดงความสัมพันธ์ระหว่างสัดส่วนสมาชิกของชุดข้อมูลคำตอบ (Label Cardinality) และความหนาแน่นของชุดข้อมูลคำตอบ (Label Density) สำหรับแต่ละชุดข้อมูลที่มีจำนวนตัวอย่าง 5000 ตัว เพื่อเปรียบเทียบลักษณะข้อมูลที่ใช้ได้ดีกับแบบจำลอง XMLC-PAO และแบบจำลองตั้งต้น..... 47

4.2 แสดงความสัมพันธ์ระหว่าง ความหนาแน่นของชุดข้อมูลคำตอบ (Label Density) และ จำนวนคำตอบของชุดข้อมูลหลายคำตอบ (Number of Labels) สำหรับแต่ละชุดข้อมูลที่มีจำนวนตัวอย่าง 5000 ตัว..... 49

4.3 แสดงความสัมพันธ์ระหว่างสัดส่วนสมาชิกของชุดข้อมูลคำตอบ (Label Cardinality) และ จำนวนคำตอบของชุดข้อมูลหลายคำตอบ (Number of Labels) สำหรับแต่ละชุดข้อมูลที่มีจำนวนตัวอย่าง 5000 ตัว..... 50

4.4 แสดงความสัมพันธ์ระหว่างสัดส่วนสมาชิกของชุดข้อมูลคำตอบ (Label Cardinality) และความหนาแน่นของชุดข้อมูลคำตอบ (Label Density) สำหรับแต่ละชุดข้อมูลเพื่อเปรียบเทียบลักษณะข้อมูลที่ใช้ได้ดีกับแบบจำลอง XMLC-PAO และแบบจำลองตั้งต้น..... 51

4.5 แสดงความสัมพันธ์ระหว่างความหนาแน่นของชุดข้อมูลคำตอบ (Label Density) และ จำนวนคำตอบของชุดข้อมูลหลายคำตอบ (Number of Labels) สำหรับแต่ละชุดข้อมูล..... 53

4.6 แสดงความสัมพันธ์ระหว่างสัดส่วนสมาชิกของชุดข้อมูลคำตอบ(Label Cardinality) และ จำนวนคำตอบของชุดข้อมูลหลายคำตอบ (Number of Labels) สำหรับแต่ละชุดข้อมูล..... 53

บทที่ 1
บทนำ

1.1 ที่มาและความสำคัญของปัญหา

ระบบการแนะนำ (Recommendation System) เป็นเครื่องมือ สำหรับแนะนำผลิตภัณฑ์หรือสินค้า (Item) ตัวอย่างเช่น ภาพยนตร์ (Movies), เพลง (Music), หนังสือ (Books), บทความงานวิจัย (Research Article) หรือ แม้แต่ กระทู้ (Online Forums) ให้แก่ผู้ใช้ (User) ที่อยู่ในระบบการใช้งาน โดยอาศัยการเทียบเคียงความคล้ายคลึงของคะแนน (Rating) ที่แสดงความสัมพันธ์ระหว่างสินค้าแต่ละชิ้นและผู้ใช้งานแต่ละคน ของสินค้าที่ผู้ใช้งานเคยซื้อ หรือใช้งานในครั้งก่อน ๆ หรือเทียบเคียงความคล้ายคลึงของคะแนน ของสินค้าที่เลือกซื้อ ระหว่างผู้ใช้ด้วยกัน ปัญหาของระบบการแนะนำ คือ การไม่มีข้อมูลตั้งต้น (Cold Start Problem) สำหรับผลิตภัณฑ์ใหม่ หรือ ผู้ใช้ใหม่ที่เพิ่งเข้ามาในระบบ

วิธีแก้ปัญหาการไม่มีข้อมูลตั้งต้นของระบบการแนะนำ แบบการคัดกรองความสัมพันธ์ (Collaborative Filtering) ที่ใช้ทั่วไปมีหลายวิธี เช่น

- แบบจำลองทางสถิติ (Statistical Model-base Approach) ซึ่งใช้สถิติศึกษาการกระจายตัวของคะแนน ในการซื้อสินค้าของผู้ใช้ก่อนหน้า มาหาความน่าจะเป็นที่จะแนะนำสินค้าชนิดใดให้แก่ผู้ใช้งานใหม่
- วิธีการเฉลี่ย (Average Approach) เป็นการใช้ค่าเฉลี่ยของคะแนน ระหว่างสินค้าแต่ละชิ้นและผู้ใช้งานแต่ละคน มาเป็นคะแนนเริ่มต้น ให้กับสินค้าแต่ละชิ้นและผู้ใช้งานใหม่แต่ละคน
- วิธีการฐานนิยม (Mode Approach) เป็นการใช้ค่าฐานนิยมของคะแนน ระหว่างสินค้าแต่ละชิ้นและผู้ใช้งานแต่ละคน มาเป็นคะแนนเริ่มต้น ให้กับสินค้าแต่ละชิ้นและผู้ใช้งานใหม่แต่ละคน

วิธีการที่กล่าวมานี้ ยังสามารถพัฒนาให้มีประสิทธิภาพ และความแม่นยำ สำหรับการใช้ในระบบแนะนำที่ดีขึ้นได้

วิธีการจำแนกประเภทข้อมูลหลายคำตอบ (Multi-label Classification) คิดค้นขึ้นครั้งแรกเพื่อใช้ในงานด้านการจัดหมวดหมู่ข้อความ (Text Categorization) และได้นำไปประยุกต์ใช้งานด้านต่าง ๆ เช่น การแยกประเภทสถานการณ์ (Semantic Scene Classification), การจำแนกอารมณ์บทเพลง (Music Emotion Categorization), การแนะนำแท็กแบบอัตโนมัติ (Automate Tag Recommendation) เป็นต้น วิธีการจำแนกประเภทข้อมูลหลายคำตอบ (Extreme Multi-label Classification) เป็นการจำแนกประเภทข้อมูลที่มีคำตอบเยอะมาก ๆ ตัวอย่างเช่น กระทู้ (Online Forums) ที่มีลักษณะเป็น คำถาม-คำตอบ ของชุมชนในระบบออนไลน์ต่าง ๆ ที่มีการตั้งคำถามเพื่อขอความคิดเห็น หรือขอคำตอบจากสมาชิกในชุมชนเดียวกัน

งานวิจัยนี้เกี่ยวข้องกับการประยุกต์ใช้ วิธีการจำแนกประเภทข้อมูลหลายคำตอบ กับระบบการแนะนำกระทู้สำหรับชุมชนออนไลน์ เพื่อทำนายกลุ่มของผู้ใช้งานที่สามารถตอบคำถามที่ตั้งขึ้นใหม่ของสมาชิกในชุมชนออนไลน์ เนื่องจากการจำแนกประเภทข้อมูลหลายคำตอบ มีจำนวนคำตอบเยอะมาก ทำให้ระยะเวลาที่ใช้ในการเรียนรู้แบบจำลองยาวนาน วัตถุประสงค์หลักของงานวิจัยชิ้นนี้ คือการศึกษาวิธีที่จะลดระยะเวลาการเรียนรู้ของแบบจำลองให้สั้นลง และเพิ่มประสิทธิภาพการทำงานของแบบจำลอง โดยการลดมิติของชุดคำตอบให้เล็กลง ก่อนจะส่งให้แบบจำลองได้ทำการเรียนรู้ โดยใช้ชุดข้อมูลกระทู้ของชุมชน สแทคโอเวอร์โฟลว์ (Stackoverflow Dataset) มาใช้ในการศึกษา

1.2 วัตถุประสงค์

1.2.1 ทำนายกลุ่มของผู้ใช้งานที่สามารถหรือสนใจ ตอบกระทู้คำถาม ที่ตั้งขึ้นมาใหม่ของสมาชิกในชุมชนออนไลน์ สแทคโอเวอร์โฟลว์

1.2.2 หาสัดส่วนที่เหมาะสมในการลดมิติของข้อมูลคำตอบ โดยใช้หลักการแปลงมิติเชิงเส้น (Principal Linear Space Transformation – PLST) เพื่อช่วยลดระยะเวลาการเรียนรู้ของแบบจำลอง

1.3 ขอบเขตงานวิจัย

1.3.1 ใช้ข้อมูลกระทู้ของชุมชนออนไลน์ สแทคโวเวอร์โฟลว์ ระหว่างปี ค.ศ. 2008-2010 จากเว็ปไซต์ https://www.kaggle.com/stackoverflow/stacksample/data

1.3.2 ปรับเปลี่ยนโครงสร้างแบบจำลองเทคนิคการเรียนรู้เชิงลึก (Deep Learning) จากงานวิจัย “Cold Start Thread Recommendation as Extreme Multi-label Classification” เพื่อใช้ในการปรับแต่งคุณสมบัติ (Feature Engineering) และมิติ (Dimension) ของข้อมูลขาเข้า

1.3.3 หาตัวดำเนินการ (Operator) ในการลดมิติของข้อมูลคำตอบ โดยใช้หลักการแปลงมิติเชิงเส้น (Principal Label Space Transformation – PLST) ซึ่งคำนวณหาตัวดำเนินการด้วยเทคนิคซิงกูลาร์ แวลู ดีคอมโพสิชัน (Singular Value Decomposition – SVD)

1.4 สมมติฐานของงานวิจัย

1.4.1 การลดมิติของข้อมูลคำตอบ สามารถลดเวลาการเรียนรู้ของแบบจำลอง โดยยังให้ประสิทธิภาพการทำงานของแบบจำลองใกล้เคียงกับ การใช้ข้อมูลคำตอบแบบไม่ลดจำนวนมิติ

1.4.2 สัดส่วนที่เหมาะสมในการลดจำนวนมิติสำหรับแต่ละชุดข้อมูล อาจจะแตกต่างกันออกไป ขึ้นอยู่กับจำนวนมิติคำตอบสำหรับแต่ละชุดข้อมูล

1.5 ประโยชน์ที่คาดว่าจะได้รับ

1.5.1 ช่วยลดระยะเวลาในการเรียนรู้ และเพิ่มประสิทธิภาพการทำงานของแบบจำลองการจำแนกประเภทข้อมูลหลายคำตอบ โดยที่การลดมิติของข้อมูลคำตอบ

1.5.2 แก้ปัญหาการขาดข้อมูลตั้งต้นสำหรับระบบแนะนำ ที่สามารถประยุกต์ใช้ได้กับชุดข้อมูลที่เป็นข้อความ เช่น การแนะนำผู้ใช้งานที่สนในจะตอบ กระทู้ถาม-ตอบ ที่เพิ่งสร้างขึ้นใหม่, การแนะนำหนังสือหรือภาพยนตร์ จากการใช้เรื่องย่อ, การแนะนำเพลงจากเนื้อเพลง เป็นต้น

1.6 นิยามศัพท์

1.6.1 ระบบการแนะนำ (Recommendation System) คือ ระบบที่ทำการเสนอผลิตภัณฑ์ที่เหมาะสมให้การสมาชิก หรือ ผู้ใช้งาน ของระบบ จากประวัติการใช้งานที่ผ่านมา

1.6.2 ผู้ใช้ หรือ ผู้ใช้งาน(User) คือ สมาชิกของระบบ หรือชุมชน โดยส่วนใหญ่มักจะเป็นลูกค้าของห้างร้าน หรือเว็ปไซต์ ที่ได้สมัครเข้าใช้งาน เพื่อจะได้มีสิทธิพิเศษในการเข้าชม หรือเลือกซื้อผลิตภัณฑ์

1.6.3 ผลิตภัณฑ์ (Item) คือ วัตถุ หรือสินค้า หรือบริการ หรือสิ่งที่ทำให้ผู้ใช้สนใช้ที่จะเข้ามาใช้งานในระบบ

1.6.4 การจำแนกประเภทข้อมูลหลายคำตอบ (Extreme Multi-label Classification) คือ การจำแนกข้อมูล ที่มีคำตอบมากกว่า 1 คำตอบ โดยปริมาณของคำตอบมีจำนวนเยอะมาก ๆ

1.6.5 หลักการแปลงมิติเชิงเส้น (Principal Linear Space Transformation – PLST) คือ หลักการที่ใช้ในการปรับเปลี่ยนมิติของข้อมูล โดยการหาตัวดำเนินการ มาคูณกับข้อมูลเดิม เพื่อปรับให้ไปอยู่ในมิติใหม่ ที่มีขนาดเล็กลง

บทที่ 2
แนวคิด ทฤษฎี และผลงานวิจัยที่เกี่ยวข้อง

2.1 ระบบการแนะนำ

ระบบการแนะนำ (Recommendation Systems) คือ ระบบที่แนะนำข้อมูล ผลิตภัณฑ์ สินค้า หรือผู้คน ซึ่งต่อไปนี้จะขอเรียกว่าผลิตภัณฑ์ (Item) ให้กับผู้ใช้งาน (User) โดยระบบจะทำการเรียนรู้ความชอบ หรือความต้องการของผู้ใช้ซึ่งอ้างอิงจากประวัติการใช้งานที่ผ่านมา ตัวอย่างการประยุกต์ใช้ระบบการแนะนำ เช่น การเลือกซื้อหนังสือ เพลง ภาพยนตร์ หรือผลิตภัณฑ์อื่น ๆ เป็นต้น หลักการของระบบการแนะนำสามารถแบ่งออกได้เป็น 3 กลุ่มใหญ่ คือ

2.1.1 การคัดกรองร่วม (Collaborative Filtering Systems)

เป็นการใช้คะแนนความสัมพันธ์ (Rating) ระหว่าง ผลิตภัณฑ์ และผู้ใช้ ซึ่งจะบ่งบอกถึงความชอบ หรือความต้องการของผู้ใช้ กับผลิตภัณฑ์นั้น โดยมีสมมติฐานว่า ผู้ใช้ที่เคยใช้ผลิตภัณฑ์หนึ่งในอดีต จะสนใจผลิตภัณฑ์ประเภทเดียวกันในอนาคต หากพิจารณาเป็นคะแนนความสัมพันธ์ ค่าของคะแนน ระหว่างผู้ใช้ และ ผลิตภัณฑ์นั้นในอดีตจะมีค่าสูง ทำให้คะแนน ระหว่างผู้ใช้ และผลิตภัณฑ์ประเภทเดียวกันกับในอดีต มีค่าสูงด้วยเช่นกัน

การคัดกรองร่วมแบ่งออกเป็น 2 ประเภทคือ

การคัดกรองร่วมโดยพิจารณาผู้ใช้ (User-based Collaborative Filtering) เป็นการพิจารณาความคล้ายคลึง ระหว่างผู้ใช้ โดยมีสมมติฐานว่า ผู้ใช้ที่มีความชอบคล้ายกันในอดีตน่าจะชอบผลิตภัณฑ์ที่เหมือนกันในอนาคต ตัวอย่างเช่น ผู้ใช้ ก และ ผู้ใช้ ข มีประวัติการซื้อหนังสือที่คล้ายกันมาก ระบบจะแนะนำหนังสือที่ผู้ใช้ ก ซื้อไปแล้ว แต่ผู้ใช้ ข ยังไม่เคยซื้อเลยให้

การคำนวณคะแนนความสัมพันธ์เพื่อแนะนำจะต้องมีตารางความสัมพันธ์ (Utility Matrix) ที่ใช้บอกคะแนนความสัมพันธ์ระหว่าง ผู้ใช้ และผลิตภัณฑ์ แต่ละชนิด ซึ่งมีลักษณะ แสดงด้านล่าง

ตารางที่ 2.1 ตารางความสัมพันธ์ (Utility Matrix)

ผู้ใช้/ผลิตภัณฑ์	ผลิตภัณฑ์ 1	ผลิตภัณฑ์ 2	ผลิตภัณฑ์ 3
ผู้ใช้ ก	1	0	4
ผู้ใช้ ข	1	2	5
ผู้ใช้ ค	2	0	0

หลักการทำงานของระบบการแนะนำแบบนี้ จะทำการหาผู้ใช้ ที่มีลักษณะเหมือนกับผู้ใช้ที่ระบบกำลังสนใจ (Active User) โดยคำนวณความเหมือนจากค่าคะแนนความสัมพันธ์ แล้วทำนายคะแนนของผลิตภัณฑ์ซึ่งผู้ใช้ที่ระบบสนใจยังไม่เคยมีประวัติการซื้อมากก่อน และแนะนำผลิตภัณฑ์ที่มีค่าคะแนนสูงกว่าคะแนนเฉลี่ยของผู้ใช้ที่ระบบสนใจ การคำนวณจะเริ่มต้นจากคำนวณค่าความเหมือน (Similarity) ระหว่างผู้ใช้แต่ละคนกับผู้ใช้ที่กำลังสนใจโดยใช้สัมประสิทธิ์สหสัมพันธ์เพียร์สัน **(Pearson Correlation Coefficient)** ดังสมการรด้านล่าง

$$c_{user} = c_{xy} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2 (y_i - \bar{y})^2}}$$

เมื่อ c_{user} คือความเหมือนของผู้ใช้ที่ระบบสนใจ กับผู้ใช้ในระบบคนอื่น ๆ จากนั้นจึงนำค่าความเหมือนที่ได้เป็นค่าน้ำหนักสำหรับคำนวณคะแนนความสัมพันธ์ (Rating) ของผลิตภัณฑ์ที่ผู้ใช้ที่ระบบสนใจแต่ยังไม่เคยใช้ โดยคำนวณค่าเฉลี่ยเลขคณิตถ่วงน้ำหนัก (Weighted Arithmetic Mean) จากคะแนนความสัมพันธ์ของผู้ใช้คนอื่น

$$rating_{item} = \frac{\sum_{j=1}^{n} rating_{item,j} * c_{user,j}}{\sum_{j=1}^{n} c_{user,j}}$$

เมื่อ $c_{user,j}$ คือ ค่าความเหมือนของผู้ใช้ที่เคยใช้ผลิตภัณฑ์นั้น กับผู้ใช้ที่ระบบสนใจแล้วนำคะแนนความสัมพันธ์ ของผลิตภัณฑ์ที่ผู้ใช้ที่ระบบสนใจแต่ยังไม่เคยใช้ มาเปรียบเทียบกับค่าเฉลี่ยของคะแนนความสัมพันธ์ของผลิตภัณฑ์ที่ผู้ใช้เคยใช้ เพื่อจะแนะนำผลิตภัณฑ์ที่มีคะแนนความสัมพันธ์สูงกว่าค่าเฉลี่ย

การคัดกรองร่วมโดยพิจารณาผลิตภัณฑ์ (Item-based Collaborative Filtering) เป็นการพิจารณาความคล้ายคลึง ระหว่างผลิตภัณฑ์ เป็นวิธีที่สร้างขึ้นเพื่อแก้ปัญหาที่พบใน *การคัดกรองร่วมโดยพิจารณาผู้ใช้* ที่จะใช้เวลาในการคำนวณนาน หากผู้ใช้มีจำนวนมาก โดยจะทำการคำนวณความคล้ายคลึงของผลิตภัณฑ์ แทนที่จะเปรียบเทียบความคล้ายคลึงระหว่างผู้ใช้ เนื่องจากจำนวนผลิตภัณฑ์น้อยกว่าจำนวนของผู้ใช้ มาก ๆ ทำให้เวลาในการคำนวณสั้นลง

การคำนวณคะแนนความสัมพันธ์จะเริ่มต้นจากการคำนวณความคล้ายคลึงกันของผลิตภัณฑ์ โดยส่วนใหญ่จะใช้วิธีการคำนวนแบบโคไซน์ (Cosine Similarity)

$$c_{item} = \frac{A.B}{\|A\|\|B\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}}$$

เมื่อ c_{item} คือความคล้ายกันสำหรับแต่ละผลิตภัณฑ์ จากนั้นจึงนำค่าความเหมือนที่ได้เป็นค่าน้ำหนักสำหรับคำนวณคะแนนความสัมพันธ์ (Rating) ของผลิตภัณฑ์ที่ผู้ใช้ที่ระบบสนใจแต่ยังไม่เคยใช้ โดยคำนวณค่าเฉลี่ยเลขคณิตถ่วงน้ำหนัก (Weighted Arithmetic Mean) จากคะแนนความสัมพันธ์ของผลิตภัณฑ์ที่ผู้ใช้ที่ระบบสนใจเคยใช้

$$rating_{item} = \frac{\sum_{j=1}^{n} rating_{item,j} * c_{item,j}}{\sum_{j=1}^{n} c_{item,j}}$$

เมื่อ $c_{item,j}$ คือ ค่าความเหมือนของผลิตภัณฑ์ที่ผู้ใช้ที่ระบบสนใจเคยใช้ ข้อด้อยของวิธีการนี้ คือ ปัญหาการขาดข้อมูลตั้งต้น (Cold Start Problem) สำหรับผู้ใช้ที่เพิ่งจะเข้ามาในระบบทำให้ไม่สามารถคำนวนคะแนนความสัมพันธ์ได้ และประสิทธิภาพการทำงานจะไม่ดีถ้ามีข้อมูลจำนวนน้อย ๆ

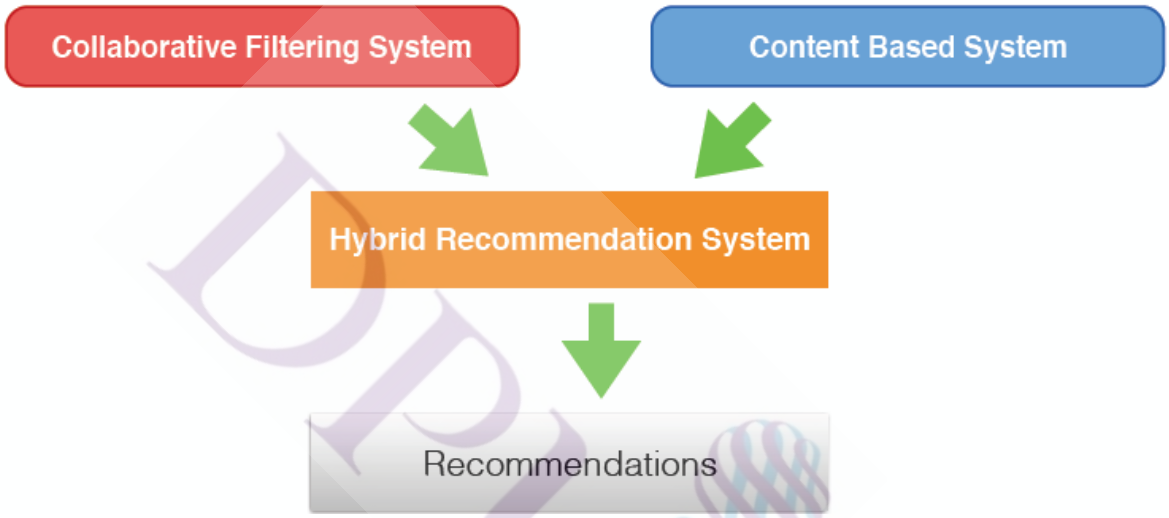
2.1.2 การแนะนำตามเนื้อหา (Content-based Recommendation Systems)

วิธีการของการคัดกรองร่วม (Collaborative Filtering) จะทำการพิจารณาเฉพาะค่าคะแนนความสัมพันธ์เท่านั้น สำหรับการแนะนำตามเนื้อหานี้ จะสร้างคุณลักษณะ (Feature) ของแต่ละผลิตภัณฑ์ เช่น ภาพยนตร์ หรือ เพลง หรือ กระทู้ โดยจะสร้างคุณลักษณะจากเนื้อหา หรือ

คุณสมบัติของผลิตภัณฑ์นั้น ๆ ออกมาในรูปของตารางความสัมพันธ์ระหว่างผู้ใช้ กับคุณลักษณะต่าง ๆ ของผลิตภัณฑ์ เพิ่มขึ้นมา และนำมาใช้ในการคำนวณค่าคะแนนความสัมพันธ์

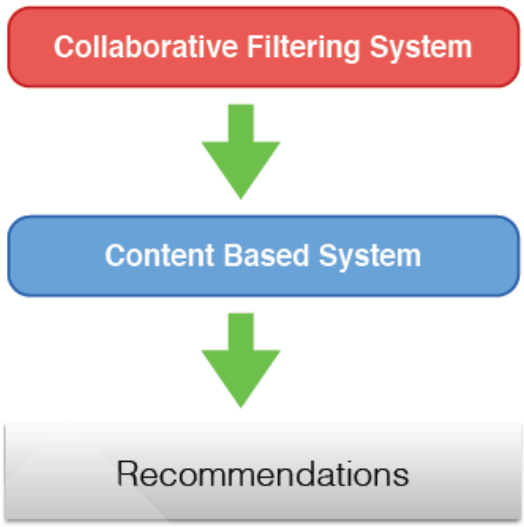
2.1.3 การแนะนำแบบไฮบริด (Hybrid Recommendation Systems)

เป็นการรวมระบบการแนะนำหลาย ๆ แนวทางเข้าด้วยกัน ตัวอย่างเช่น การแนะนำแบบไฮบริดถ่วงน้ำหนัก (Weighted Method) เป็นการผสานการคำนวนคะแนนความสัมพันธ์ จากระบบการแนะนำแบบอื่น ๆ โดยใช้การถ่วงน้ำหนัก ดังแสดงในแผนผังภาพที่ 2.1



ภาพที่ 2.1 แผนผังแสดงแนวคิดระบบการแนะนำแบบไฮบริดถ่วงน้ำหนัก

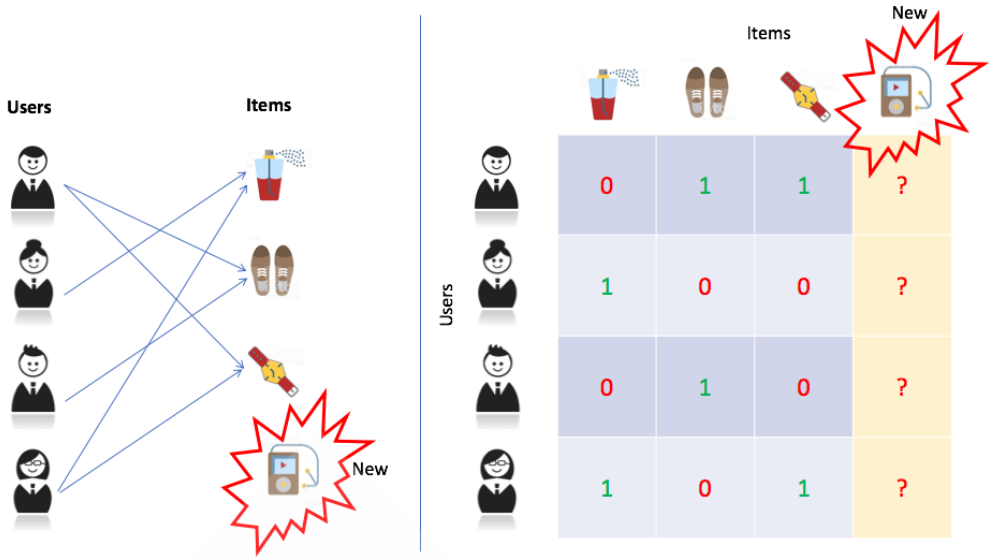
การแนะนำแบบไฮบริดตามลำดับขั้น (Cascade Method) เป็นการผสารการคำนวณคะแนนความสัมพันธ์ จากระบบการแนะนำแบบอื่น ๆ แบบตามลำดับเป็นขั้น ดังแผนผังแสดงแนวคิดในภาพที่ 2.2



ภาพที่ 2.2 แผนผังแสดงแนวคิดระบบการแนะนำแบบไฮบริดตามลำดับขั้น

2.2 ปัญหาการขาดข้อมูลตั้งต้น (Cold Start Recommendation System Problem)

ส่วนประกอบที่สำคัญสำหรับระบบการแนะนำ มี 2 ส่วน ได้แก่ ผู้ใช้ (User) และผลิตภัณฑ์ (Item) ซึ่งความสัมพันธ์ระหว่างผู้ใช้ และ ผลิตภัณฑ์ อยู่ในรูปแบบกราฟสองฝั่ง (Bipartite Graph) มีลักษณะดังภาพที่ 2.3 (ก) โดยเส้นเชื่อมความสัมพันธ์แบบมีทิศทาง จะชี้จากด้านที่เป็นผู้ใช้ไปยังด้านที่เป็นผลิตภัณฑ์เสมอ ตารางความสัมพันธ์ระหว่างผู้ใช้ และผลิตภัณฑ์ มีลักษณะดังแสดงในภาพที่ 2.3 (ข) หากมีส่วนหนึ่ง ส่วนใดที่เพิ่งจะเข้ามาในระบบใหม่ เส้นเชื่อมความสัมพันธ์ระหว่างผู้ใช้ และ ผลิตภัณฑ์ จะไม่ปรากฏ ทำให้ผู้ใช้ที่เพิ่งมาในระบบไม่ได้รับการแนะนำ หรือผลิตภัณฑ์ที่เพิ่งเข้ามาในระบบไม่ถูกแนะนำ



(ก) (ข)

ภาพที่ 2.3 (ก) แสดงความสัมพันธ์ระหว่างผู้ใช้ (Users) และผลิตภัณฑ์ (Items)แบบกราฟสองฝั่ง (Bipartite Graph)
(ข) ตารางความสัมพันธ์ระหว่างผู้ใช้แต่ละคน และผลิตภัณฑ์แต่ละอย่าง

จากภาพผลิตภัณฑ์หมายเลข 4 เป็นผลิตภัณฑ์ใหม่ที่เพิ่งเข้ามาในระบบ จึงยังไม่มีค่าคะแนนความสัมพันธ์ ทำให้ผลิตภัณฑ์นี้จะไม่ได้รับการแนะนำในระบบการแนะนำ

2.3 การจำแนกประเภทข้อมูลหลายคำตอบ (Extreme Multi-label Classification - XMLC)

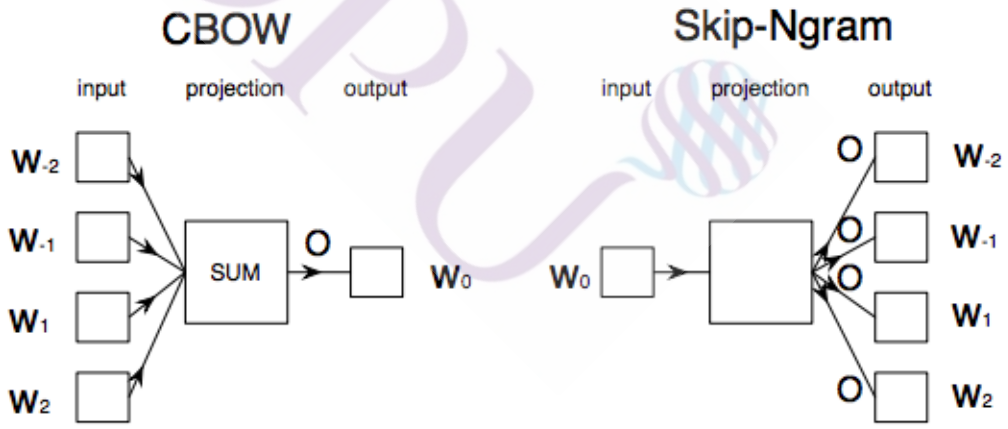
การจำแนกประเภทข้อมูลหลายคำตอบ เป็นงานที่เกี่ยวข้องกับจำแนกข้อมูลที่มีหลายคำตอบ โดยการเลือกกลุ่มของผลิตภัณฑ์จากคำตอบที่มีจำนวนเยอะมาก ๆ ความแตกต่างระหว่างการจำแนกประเภทหลายคำตอบ (Multi-label Classification) กับ การจำแนกประเภทที่มี 2 คำตอบ (Binary Classification) หรือ การจำแนกประเภทที่มีมากกว่า 2 คำตอบ (Multi-class Classification) อยู่ตรงที่ การจำแนกประเภทที่มีมากกว่า 2 คำตอบ ระบบจะเลือกคำตอบใดคำตอบหนึ่งในหลายคำตอบ มาแสดงผลเพียงคำตอบเดียว ขณะที่การจำแนกประเภทหลายคำตอบสามารถเทียบเคียงคำตอบหนึ่ง ไปยังอีกคำตอบหนึ่งได้ ทำให้ระบบแนะนำคำตอบออกมามากกว่าหนึ่งคำตอบ ตัวอย่างเช่น การเมือง และ ทำเนียบขาว สามารถเป็นคำตอบของหัวข้อข่าวเดียวกันได้ เป็นต้น

สำหรับเงื่อนไขเช่นนี้ ตัวอย่างข้อมูลหนึ่งตัว สามารถมองได้ว่าเป็นคู่อันดับ (x,y) โดยที่ x เป็นเวกเตอร์ลักษณะ (Feature Vector) สำหรับผลิตภัณฑ์หนึ่ง ส่วน y เป็นเวกเตอร์คำตอบ (Label Vector) ตัวอย่างเช่น $y \in \{0,1\}^L$ โดยที่ L เป็นจำนวนของคำตอบ

2.4 การแทนค่าคำ (Word Embedding)

การแทนค่าคำ เป็นวิธีการที่ได้รับความนิยมมากสำหรับการสร้างตัวแทนของเอกสาร โดยคำที่นำมาใช้เป็นตัวแทนสามารถที่จะเก็บรายละเอียดของเนื้อหาที่อยู่ในเอกสาร, ความหมาย, ความคล้ายคลึงกันของประโยค และความสัมพันธ์ระหว่างคำอื่น ๆ

เวิร์ดทูเวก (Word2Vec) เป็นวิธีที่นิยมอีกวิธีสำหรับ การแทนค่าคำ (Word Embedding) โดยการแปลงคำที่เป็นตัวแทนของเอกสาร ให้อยู่ในรูปของเวกเตอร์ (Tomas Mikolov, 2013) โดยอาศัยหนึ่งในแบบจำลองทางโครงข่ายประสาท “ซีโบว์” (Continuous Bag of Words - CBOW) และ “สคิปแกรม” (Skip Gram) ทั้งสองวิธีจะทำการศึกษาใจความสำคัญของคำรอบข้างเพื่อสรุปเป็นใจความสำคัญของข้อความ (Wang Ling, 2015) (ภาพที่ 2.4)



ภาพที่ 2.4 แผนผังวิธีทางโครงข่ายประสาท แบบซีโบว์ และ สคิปแกรม

สคิปแกรม มีหลักการที่จะใช้คำที่เป็นศูนย์กลางเพื่อทำนายคำแวดล้อม ส่วนซีโบว์ จะใช้คำแวดล้อมเพื่อทำนายคำที่เป็นศูนย์กลาง ซึ่งจะเห็นได้ว่าทั้งสองวิธีไม่ได้สนใจลำดับของคำแวดล้อมในการทำนายคำตอบ ทำให้ทั้งสองวิธีการนี้อาจจะให้ตัวแทนของเอกสารที่คล้ายคลึงกันสำหรับกลุ่มของคำที่คล้าย ๆ กัน แม้ความหมายจะแตกต่างกัน

ต่อมาจึงได้มีการพัฒนาหลักการโกลว (GloVe - Jeffrey Pennington, 2014) เพื่อการแทนค่าคำขึ้น โดยใช้สถิติของคำที่ปรากฏอยู่ในคลังข้อความ (Corpus) เป็นแหล่งข้อมูลหลักสำหรับการเรียนรู้แบบไม่มีตัวอย่าง (Unsupervised Learning) ของการแทนค่าคำ แล้วคำนวณความน่าจะเป็นของคำที่จะใช้ร่วมกัน ดังตัวอย่างในตารางที่ 2.1

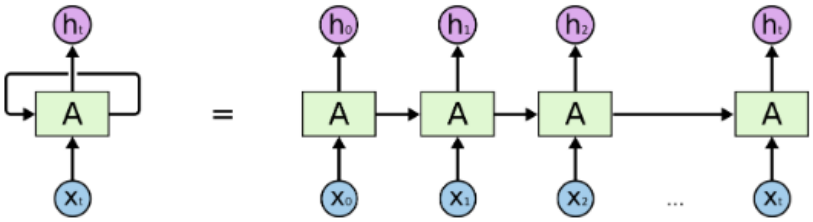
จากความน่าจะเป็นของคำแต่ละคำในตาราง คือจุดเริ่มต้นสำหรับการเรียนรู้ที่จะสร้างเวกเตอร์คำ (Word Vector) ความน่าจะเป็นของคำที่จะเกิดร่วมกัน สร้างเป็นความน่าจะเป็นของคำคำใหม่ขึ้น ทำให้คำทุกคำที่อยู่ในคลังข้อมูลมีความสัมพันธ์กัน และสามารถนำมาใช้ในการเรียนรู้เพื่อทำนายคำที่จะใช้แทนใจความสำคัญออกมา

ตารางที่ 2.2 ความน่าจะเป็นของคำที่จะเกิดร่วมกัน สำหรับคำว่า “ice” และ “stream” ค่าความน่าจะเป็นที่มากที่สุดจะสัมพันธ์กับคุณสมบัติของ “ice” ส่วนค่าความน่าจะเป็นที่น้อยที่สุดจะสัมพันธ์กับคุณสมบัติของ “stream”

Probability and Ratio	$k = solid$	$k = gas$	$k = water$	$k = fashion$
$P(k\|ice)$	1.9×10^{-4}	6.6×10^{-5}	3.0×10^{-3}	1.7×10^{-5}
$P(k\|steam)$	2.2×10^{-5}	7.8×10^{-4}	2.2×10^{-3}	1.8×10^{-5}
$P(k\|ice)/P(k\|steam)$	8.9	8.5×10^{-2}	1.36	0.96

2.5 โครงข่ายประสาทแบบวนรอบ (Recurrent Neural Network)

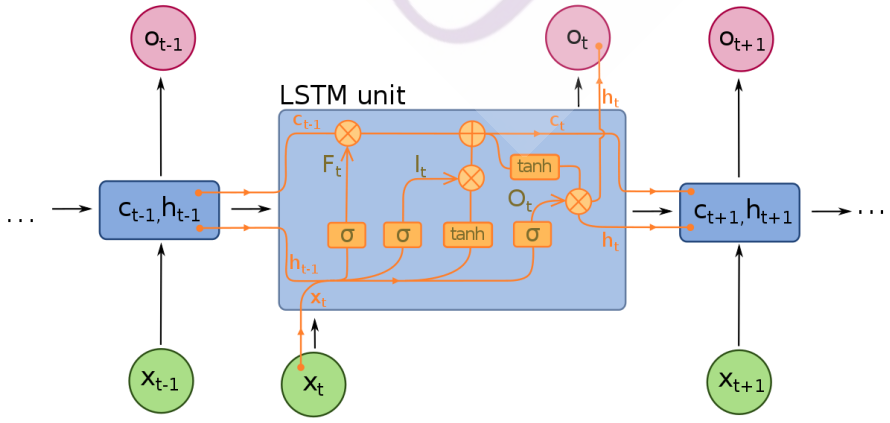
โครงข่ายประสาทแบบวนรอบ (Recurrent Neural Network – RNN) คือประเภทหนึ่งของ โครงข่ายประสาทเทียม (Artificial Neural Network) ซึ่งเป็นโครงข่ายที่มีหน่วยความจำภายในสำหรับใช้ประมวลผลข้อมูลขาเข้าที่มีลำดับ (Sequential Input) โครงข่ายประสาทประเภทที่เป็นที่รู้จักกันดีได้แก่ โครงข่ายประสาทแบบวนรอบที่มีหน่วยความจำสั้น-ยาว (Long-Shot Term Memory – LSTM) และโครงข่ายประสาทแบบวนรอบที่มีหน่วยความจำแบบย่อ (Gate Recurrent Unit – GRU)



ภาพที่ 2.5 แผนผังโครงข่ายประสาทแบบวนรอบ

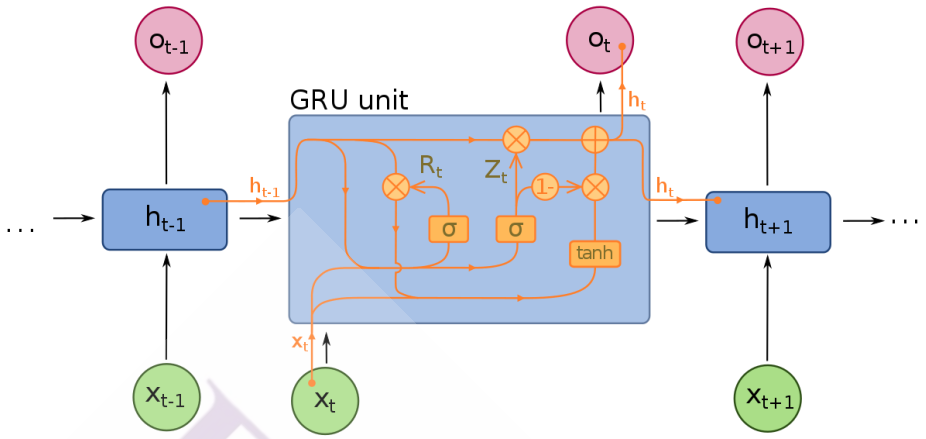
โครงข่ายประสาททั้งสองแบบนี้ จะมีหน่วยความจำภายในที่สร้างขึ้นมาเพื่อแก้ไขการลดน้อยลงของความคลาดเคลื่อนที่เกิดขึ้น (Vanishing Gradient Problem) กับโครงข่ายประสาทแบบวนรอบแบบเดิม ซึ่งไม่มีหน่วยความจำทำให้ระบบลืมที่จะพิจารณาลำดับของข้อมูลขาเข้าที่ก่อนหน้านาน ๆ เช่น ข้อความที่ค่อนข้างยาว ระบบจะพิจารณาเฉพาะคำรอบข้างของข้อมูลขาเข้าที่กำลังสนใจ จนอาจจะทำให้ความหมายของข้อความเพี้ยนไปได้ ภายในโครงข่ายประสาทแบบวนรอบที่มีหน่วยความจำสั้น-ยาว (LSTM) และ โครงข่ายปสาทแบบวนรอบที่มีหน่วยความจำแบบย่อ (GRU) จะมีส่วนประกอบที่เรียกว่า ประตู (Gate)

โครงข่ายประสาทแบบวนรอบที่มีหน่วยความจำสั้น-ยาว (LSTM) จะมีประตูการลืม (Forget Gate) อยู่ภายใน ทำให้ค่าความคลาดเคลื่อนที่เกิดขึ้นสามารถกลับเข้ามาใช้ในการเรียนรู้ของระบบ ช่วยให้ระบบไม่ลืมสิ่งที่ผ่านมาแล้วนาน ๆ



ภาพที่ 2.6 แผนผังโครงข่ายประสาทแบบวนรอบที่มีหน่วยความจำสั้นยาว

โครงข่ายประสาทแบบวนรอบที่มีหน่วยความจำแบบย่อ (GRU) เป็นการลดความซับซ้อนของโครงข่ายประสาทแบบก่อนหน้าลง เพื่อให้ระบบทำงานได้เร็วขึ้นและมีประสิทธิภาพเท่าเทียมกัน



ภาพที่ 2.7 แผนผังโครงข่ายประสาทแบบวนรอบที่มีหน่วยความจำแบบย่อ

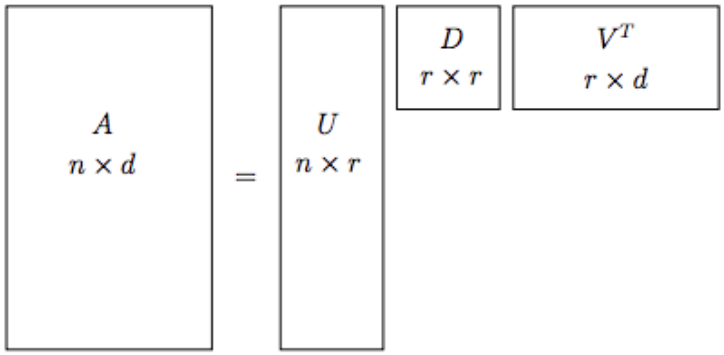
โครงข่ายประสาทแบบวนรอบสองทิศทาง (Bi-directional RNNs) เป็นการใช้ลำดับของข้อมูลขาเข้า ที่เดินจากอดีตไปสู่อนาคต หรือ จากซ้ายไปขวา และเดินจากอนาคตไปสู่อดีต หรือจากขวาไปซ้าย โดยเอาคำตอบจาก โครงข่ายประสาทแบบวนรอบในแต่ละทิศทางมาเชื่อมต่อกัน เพื่อให้ลำดับของข้อมูลขาเข้าถูกต้องมากขึ้น มักจะใช้ในงานการประมวลผลด้านภาษา (Natural Language Processing – NLP)

2.6 เทคนิคซิงกูลาร์ แวลู ดีคอมโพสิชัน (Singular Value Decomposition – SVD)

เทคนิคซิงกูลาร์ แวลู ดีคอมโพสิชัน เป็นการแยกส่วนประกอบของเมทริกซ์ (Matrix) หนึ่งออกมาในรูปผลคูณของเมทริกซ์ 3 ส่วน

$$A = UDV^T$$

โดยที่ U และ V เป็นเมทริกซ์ที่มีคอลัมน์ตั้งฉากกัน (Orthonomal) ส่วน D เป็นเมทริกซ์แนวทแยง หลักการนี้ได้นำไปประยุกต์ใช้กับงานต่าง ๆ มากมาย ตัวอย่างเช่น การลดขนาดมิติของเมทริกซ์ให้เล็กลง



ภาพที่ 2.8 เทคนิคซิงกูลาร์ แวลู ดีคอมโพสิชัน สำหรับเมทริกซ์ A ที่มีขนาด $n \times d$

2.7 งานวิจัยที่เกี่ยวข้อง

งานวิจัยชิ้นนี้เกี่ยวข้องกับการประยุกต์ใช้การจำแนกประเภทข้อมูลหลายคำตอบ (Extreme Multi-label Classification - XMLC) มาแก้ปัญหาการขาดข้อมูลตั้งต้นของระบบการแนะนำ (Cold Recommendation System Problem) โดยมีจุดประสงค์เพื่อแนะนำกระทู้ใหม่ในชุมชนออนไลน์ สแทคโอเวอร์โฟลว์ (Stackoverflow Dataset) ให้แก่ผู้ใช้ที่มีน่าจะมีความสนใจเนื้อหาของกระทู้นั้น

แนวคิดที่นำมาใช้เป็นการเปลี่ยนมุมมองของการมองปัญหาให้เป็นการเรียนรู้เพื่อจำแนกข้อมูลหลายระดับ แล้วจึงนำไปใช้เป็นข้อมูลตั้งต้นของระบบการแนะนำ (Kishaloy Halder, 2018) โดยการประยุกต์ใช้ การจำแนกประเภทข้อมูลหลายคำตอบมาใช้กับการจำแนกประเภทข้อความ (Text Classification) ที่สามารถแยกแยะออกได้เป็นหลายประเภท การใช้เทคนิคการเรียนรู้เชิงลึก (Deep Learning) มาช่วยสามารถทำให้แบบจำลองเข้าใจ และรับมือกับคำตอบจำนวนเยอะมาก ๆ ได้ดี (Jingzhou Liu, 2017)

แม้ว่าวิธีการจำแนกประเภทข้อมูลหลายคำตอบ จะสามารถนำมาช่วยคัดกรองผู้ใช้ มาใส่ในระบบเพื่อแก้ปัญหาการขาดข้อมูลตั้งต้นของระบบการแนะนำได้ แต่ยังมีข้อเสียที่ชุดคำตอบซึ่งหรือผู้ใช้ในระบบ มีขนาดที่ใหญ่มาก ๆ และมีข้อมูลอยู่แบบเบาบาง (Sparse) ทำให้การเรียนรู้ใช้เวลานานความเป็นไปได้ที่จะรับมือกับข้อมูลแบบนี้คือการแปลงชุดคำตอบให้ไปอยู่มิติที่เล็กลงโดยใช้หลักการแปลงมิติเชิงเส้น (Principal Linear Space Transformation – PLST) (F. Tai, 2010) ซึ่งเป็นวิธีที่ง่ายและมีประสิทธิภาพ แล้วจึงนำชุดคำตอบนี้ไปใช้สร้างแบบจำลอง โดยตัวดำเนินการที่ใช้ในการแปลงมิติ สามารถคำนวณได้โดยเทคนิคซิงกูลาร์ แวลู ดีคอมโพสิชัน (Singular Value Decomposition – SVD)

การแบ่งผู้ใช้ออกเป็นกลุ่มตามแต่ความสนใจในมิติแฝง (Latent Space) โดยมีสมมติฐานว่าผู้ใช้ที่อยู่ในกลุ่มเดียวกันจะมีความชอบที่คล้ายคลึงกัน ทำให้พบความคล้ายคลึงของใจความในข้อความที่แต่ละกลุ่มสนใจ ซึ่งสามารถนำมาประยุกต์ในการแบ่งประเภทใจความของข้อความ (Cluster Sensitive Attention – CSA) (Bin Xu, 2012)

บทที่ 3
ระเบียบวิธีวิจัย

งานวิจัยนี้เป็นการประยุกต์ใช้แบบจำลองการเรียนรู้เชิงลึก (Deep Learning Model) กับการจำแนกประเภทข้อมูลหลายคำตอบ (Extreme Multi-label Classification) ซึ่งมุ่งหมายที่จะศึกษาการลดจำนวนมิติของข้อมูลคำตอบ เพื่อลดระยะเวลาในการเรียนรู้ และเพิ่มประสิทธิภาพการทำงานของแบบจำลอง รายละเอียดระเบียบวิธีวิจัยได้แสดงไว้ในบทนี้ โดยจะเริ่มต้นจากลักษณะของชุดข้อมูล, การเตรียมข้อมูล, การคำนวณตัวดำเนินการแปลงมิติเชิงเส้น, โครงสร้างของแบบจำลอง, การวัดประสิทธิภาพของแบบจำลอง, และแนวทางการวิจัย

3.1 ลักษณะของชุดข้อมูล

การจัดเตรียมข้อมูลเพื่อให้อยู่ในรูปแบบที่จะส่งเข้าไปในแบบจำลอง ถือเป็นสิ่งสำคัญมาก ข้อมูลที่เลือกมาใช้ในงานวิจัยนี้คือ ชุดข้อมูลกระทู้ในชุมชนออนไลน์ สแทคโอเวอร์โฟลว์ (Stackoverflow dataset) ระหว่างปี ค.ศ. 2008-2010 จากเว็ปไซต์ *https://www.kaggle.com/stackoverflow/stacksample/data* โดยชุดข้อมูลจะแบ่งออกเป็น 3 ส่วนคือ

3.1.1. ชุดข้อมูลคำถาม ซึ่งอยู่ในไฟล์ Questions.csv.zip ชุดข้อมูลนี้มีลักษณะเป็น ตารางที่มี 7 คอลัมน์ คือ (ดังแสดงในภาพที่ 3.1)

1) Id คือ หมายเลขข้อความ
2) OwerUserId คือ เลขประจำตัวของผู้ใช้ที่ตั้งคำถาม
3) CreationDate คือ วันที่ข้อความคำถามถูกสร้าง
4) ClosedDate คือ วันที่ข้อความคำถามถูกปิดไป
5) Score คือ คะแนนความนิยมของคำถามนั้น
6) Title คือ ชื่อหัวข้อคำถาม
7) Body คือ เนื้อหาของคำถาม ดังตัวอย่างในภาพที่ 3.2

3.1.2. ชุดข้อมูลคำตอบ ซึ่งอยู่ในไฟล์ Answers.csv.zip ชุดข้อมูลนี้มีลักษณะเป็น ตารางที่มี 6 คอลัมน์ คือ (ดังแสดงในภาพที่ 3.3)

1) Id คือ หมายเลขข้อความ
2) OwerUserId คือ เลขประจำตัวของผู้ใช้ที่ตอบ
3) CreationDate คือ วันที่ข้อความคำตอบถูกสร้าง
4) ParentId คือ หมายเลขข้อความคำถาม (Id ในชุดข้อมูลคำถาม)
5) Score คือ คะแนนความนิยมของคำตอบนั้น
6) Body คือ เนื้อหาของคำตอบ ดังตัวอย่างในภาพที่ 3.4

	Id	OwnerUserId	CreationDate	ClosedDate	Score	Title	Body
0	80	26.0	2008-08-01T13:57:07Z	NaN	26	SQLStatement.execute() - multiple queries in o...	<p>I've written a database generation script i...
1	90	58.0	2008-08-01T14:41:24Z	2012-12-26T03:45:49Z	144	Good branching and merging tutorials for Torto...	<p>Are there any really good tutorials explain...
2	120	83.0	2008-08-01T15:50:08Z	NaN	21	ASP.NET Site Maps	<p>Has anyone got experience creating ...
3	180	2089740.0	2008-08-01T18:42:19Z	NaN	53	Function for creating color wheels	<p>This is something I've pseudo-solved many t...
4	260	91.0	2008-08-01T23:22:08Z	NaN	49	Adding scripting functionality to .NET applica...	<p>I have a little game written in C#. It uses...

ภาพที่ 3.1 ตัวอย่างของชุดข้อมูลคำถาม

```
"<p>Has anyone got experience creating <strong>SQL-based ASP.NET</strong> site-map providers?</p>\n\n<p>I've got the
default XML file <code>web.sitemap</code> working properly with my Menu and <strong>SiteMapPath</strong> controls, bu
t I'll need a way for the users of my site to create and modify pages dynamically.</p>\n\n<p>I need to tie page viewi
ng permissions into the standard <code>ASP.NET</code> membership system as well.</p>\n"
```

ภาพที่ 3.2 ตัวอย่างข้อความในคอลัมน์ Body ของชุดข้อมูลคำถาม

	Id	OwnerUserId	CreationDate	ParentId	Score	Body
0	92	61.0	2008-08-01T14:45:37Z	90	13	<p><a href="http://svnbook.red-bean.com/">Vers...
1	124	26.0	2008-08-01T16:09:47Z	80	12	<p>I wound up using this. It is a kind of a ha...
2	199	50.0	2008-08-01T19:36:46Z	180	1	<p>I've read somewhere the human eye can't dis...
3	269	91.0	2008-08-01T23:49:57Z	260	4	<p>Yes, I thought about that, but I soon figur...
4	307	49.0	2008-08-02T01:49:46Z	260	28	<p><a href="http://www.codeproject.com/Article...

ภาพที่ 3.3 ตัวอย่างของชุดข้อมูลคำตอบ

" I've read somewhere the human eye can't distinguish between less than 4 values apart. so This is something to kee
p in mind. The following algorithm does not compensate for this. \r\n\r\n I'm not sure this is exactly what you
want, but this is one way to randomly generate non-repeating color values: \r\n\r\n (beware, inconsistent pseudo
-code ahead) \r\n\r\n<pre><code>//colors entered as 0-255 [R, G, B]
colors = []; //holds final colors to be use
d
rand = new Random();

//assumes n is less than 16,777,216
randomGen(int n){
  while (len(colors) &l
t; n){
  //generate a random number between 0,255 for each color
  newRed = rand.next(256);
  ne
wGreen = rand.next(256);
  newBlue = rand.next(256);
  temp = [newRed, newGreen, newBlue];
  //o
nly adds new colors to the array
  if temp not in colors {
  colors.append(temp);
  }
}
}
</code></pre>\r\n\r\n One way you could optimize this for better visibility would be to compare the dista
nce between each new color and all the colors in the array: \r\n\r\n<pre><code>for item in color{
  itemSq =
(item[0]^2 + item[1]^2 + item[2]^2])^(.5);
  tempSq = (temp[0]^2 + temp[1]^2 + temp[2]^2])^(.5);
  dist = ite
mSq - tempSq;
  dist = abs(dist);
}
//NUMBER can be your chosen distance apart.
if dist < NUMBER and
temp not in colors {
  colors.append(temp);
}
</code></pre>\r\n\r\n But this approach would significantly
slow down your algorithm. \r\n\r\n Another way would be to scrap the randomness and systematically go through ev
ery 4 values and add a color to an array in the above example. "

ภาพที่ 3.4 ตัวอย่างข้อความในคอลัมน์ Body ของชุดข้อมูลคำตอบ

3.1.3. ชุดข้อมูลแท็ก ซึ่งอยู่ในไฟล์ Tags.csv.zip ซึ่งประกอบด้วย 2 คอลัมน์ ดังแสดงในภาพที่ 3.5

1) Id คือ หมายเลขข้อความ
2) Tag คือ ชื่อแท็ก ของแต่ละข้อความ

ซึ่งชุดข้อมูลนี้ไม่ได้นำมาใช้ในงานวิจัย

	Id	Tag
0	80	flex
1	80	actionscript-3
2	80	air
3	90	svn
4	90	tortoisesvn

ภาพที่ 3.5 ตัวอย่างของชุดข้อมูลแท็ก

3.2 การเตรียมข้อมูล

หลังจากตรวจสอบข้อมูลแต่ละชุด พบว่าข้อมูลที่สามารถนำมาใช้ในงานวิจัยได้ คือ ชุดข้อมูลคำถาม และชุดข้อมูลคำตอบ ซึ่งจำเป็นต้องมีการปรับแต่งให้เหมาะสมที่จะนำไปใช้งาน ตามขั้นตอนดังนี้

3.2.1 จัดการข้อความในคอลัมน์ Body ของชุดข้อมูลคำถาม

จากขอบเขตงานวิจัย ที่สนใจจะแนะนำข้อความกระทู้หรือข้อความที่ถูกสร้างขึ้นใหม่ให้กับผู้ใช้ หรือสมาชิกที่มีความสนใจจะตอบกระทู้นั้น กระทู้ที่ถูกสร้างขึ้นในชุดข้อมูลของชุมชนออนไลน์สแทคโอเวอร์โฟลว์ บันทึกอยู่ในชุดข้อมูลคำถาม สังเกตได้จากชุดข้อมูลนี้มีหัวข้อชื่อคำถาม (Title) อยู่ในชุดข้อมูล

ขั้นตอนแรกของการจัดเตรียมข้อมูล เริ่มจากการจัดการข้อความที่เป็นเนื้อหาของคำถามในคอลัมน์ Body จากตัวอย่างข้อความส่วนนี้ในภาพที่ 3.2 เห็นได้ว่า มีส่วนที่ไม่ใช้ข้อความรวมอยู่ด้วย ซึ่งจำเป็นต้องตัดออกไป เพื่อไม่ให้เนื้อหาของข้อความผิดเพี้ยน โดยการใช้ชุดคำสั่งสำเร็จรูป (library) ของภาษาไพตอน (Python) ที่ชื่อว่า “re” ตัวอย่าง การเขียนฟังก์ชันที่ให้แก้ปัญหานี้ แสดงไว้ในภาพที่ 3.6 เมื่อผ่านการแปลงข้อความแล้วผลลัพธ์จากข้อความในภาพที่ 3.2 จะกลายเป็นดังนี้

"Has anyone got experience creating SQL-based ASP.NET site-map providers? I've got the default XML file web.sitemap working properly with my Menu and SiteMapPath controls, but I'll need a way for the users of my site to create and modify pages dynamically. I need to tie page viewing permissions into the standard ASP.NET membership system as well."

```
import re

TAG_RE = re.compile(r'<[^>]+>')
n_RE = re.compile(r'\n')

def remove_tags(text):
    output = TAG_RE.sub('', text)
    output = n_RE.sub(' ', output)
    return output

remove_tags(question.Body[2])
```

ภาพที่ 3.6 ตัวอย่างการเขียนฟังก์ชันสำหรับใช้ตัดส่วนที่ไม่จำเป็นของข้อความ ในคอลัมน์ Body ของชุดข้อมูลคำถาม

สร้างชุดข้อมูลที่มีความสัมพันธ์ระหว่างคำถาม กับผู้ใช้ที่เข้ามาให้คำตอบ

สำหรับแต่ละคำถามในชุดข้อมูลคำถาม จะมีผู้ใช้งานเข้ามาตอบแตกต่างกัน โดยเลขประจำตัวผู้ใช้ที่มาตอบจะอยู่ในชุดข้อมูลคำตอบคอลัมน์ Parent ID ในชุดข้อมูลคำตอบ ที่บอกหมายเลขข้อความคำถาม สำหรับข้อมูลคำตอบที่ผู้ใช้เข้าไปตอบคำถามเดียวกัน จะพบว่ามีเลข Parent ID ที่เหมือนกัน

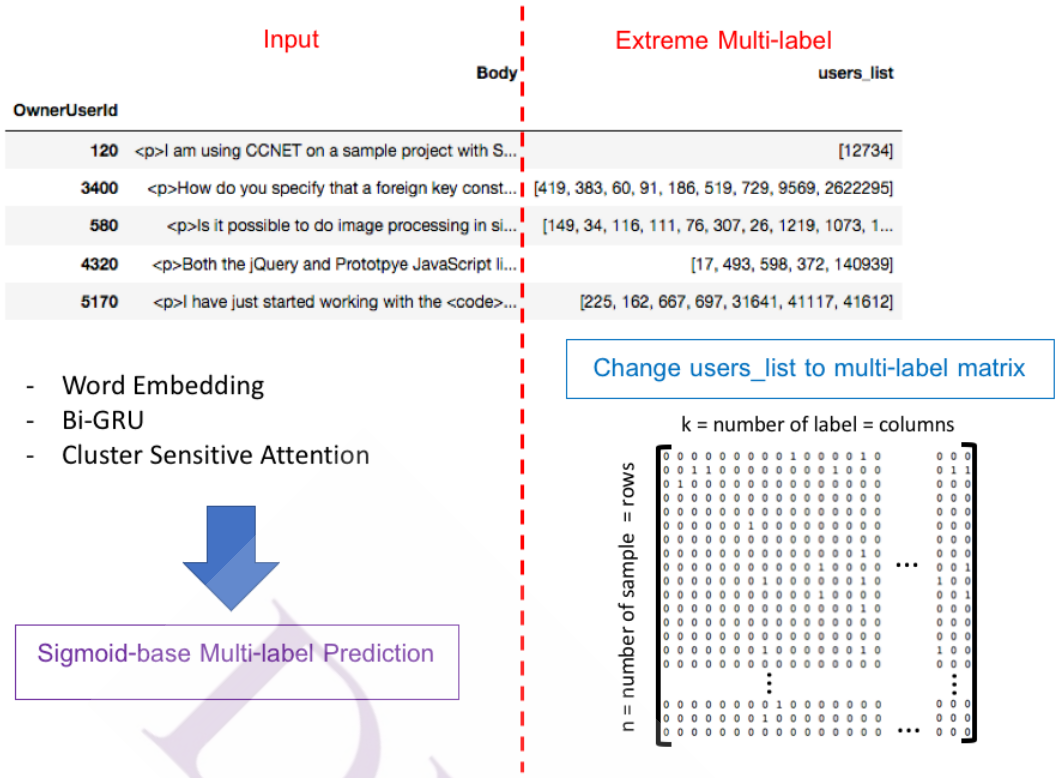
เนื่องจากข้อมูลที่ต้องการไปใช้สร้างแบบจำลองจะประกอบด้วยข้อความคำถามเป็นข้อมูลขาเข้า (Input) และรายการของผู้ใช้ที่เกี่ยวข้องกับข้อมูลขาเข้าสำหรับใช้เป็นชุดคำตอบ (Label) ขั้นตอนนี้ จะทำการสร้างชุดข้อมูลให้อยู่ในรูปแบบที่แบบจำลองต้องการ โดยการจับคู่คอลัมน์ Body ในชุดข้อมูลคำถาม ซึ่งเป็นคำถามที่แตกต่างกัน กับรายการ OwnerUserId ซึ่งเป็นเลขประจำตัวของผู้ใช้ที่เข้ามาตอบคำถามนั้นจากชุดข้อมูลคำตอบ ตัวอย่างชุดข้อมูลที่สร้างขึ้น ที่พร้อมส่งเข้าไปให้แบบจำลองได้เรียนรู้ มีลักษณะดังภาพที่ 3.7 (ก)

จากภาพที่ 3.7 (ข) คอลัมน์ Body จะถูกส่งผ่านเข้าไปยังแบบจำลองการเรียนรู้เชิงลึกโดยผ่านส่วนของการแทนค่าคำ (Word Embedding) ต่อไปยังส่วนของโครงข่ายประสาทวนรอบสองทิศทางที่มีหน่วยความจำแบบย่อ (Bi-GRU) และส่วนของการจัดกลุ่มข้อความตามใจความสำคัญ (Cluster Sensitive Attention) เพื่อสกัดคุณลักษณะของข้อมูลขาเข้า ก่อนที่จะส่งต่อไปยังส่วนการเรียนรู้ ซึ่งจะได้กล่าวถึงรายละเอียดต่อไปในหัวข้อที่ 3.4

สำหรับคอลัมน์ users_list เป็นส่วนของข้อมูลคำตอบ หรือรายการของผู้ใช้ที่จะได้รับการแนะนำให้ตอบแต่ละคำถาม โดยแต่ละคำถาม (ข้อมูลขาเข้า) สามารถแนะนำผู้ใช้ได้มากกว่าหนึ่งคน และจะถูกนำมาจัดให้อยู่ในรูปแบบของเมทริกซ์ที่จำนวนหลัก (number of columns) เท่ากับจำนวนผู้ใช้ทั้งหมดของระบบ และจำนวนแถว (number of rows) จะเท่ากับจำนวนของข้อความคำถาม โดยแต่ละแถวจะแสดงค่าของผู้ใช้ที่จะได้รับการแนะนำให้ตอบคำถามเป็น 1 และผู้ใช้ที่ไม่ควรแนะนำ จะมีค่าเป็น 0 สำหรับคำถามแต่ละข้อ ดังภาพที่ 3.7 (ข)

OwnerUserId	Body	users_list
120	<p>I am using CCNET on a sample project with S...	[12734]
3400	<p>How do you specify that a foreign key const...	[419, 383, 60, 91, 186, 519, 729, 9569, 2622295]
580	<p>Is it possible to do image processing in si...	[149, 34, 116, 111, 76, 307, 26, 1219, 1073, 1...
4320	<p>Both the jQuery and Prototpye JavaScript li...	[17, 493, 598, 372, 140939]
5170	<p>I have just started working with the <code>...	[225, 162, 667, 697, 31641, 41117, 41612]

(ก) ตัวอย่างข้อมูลขาเข้า (input) แสดงอยู่ในคอลัมน์ Body และชุดข้อมูลคำตอบ (label) แสดงอยู่ในคอลัมน์ users_list

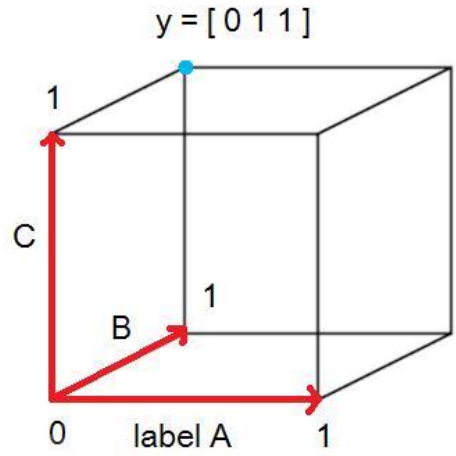


(ข) รายละเอียด การจัดการข้อมูลเข้า (input) ในโครงข่ายประสาท และการจัดการชุดข้อมูลคำตอบ (output) ที่มีหลายคำตอบ

ภาพที่ 3.7 ข้อมูลขาเข้า และ ข้อมูลคำตอบ สำหรับการเรียนรู้ของแบบจำลอง

3.3 การคำนวณตัวดำเนินการแปลงมิติเชิงเส้น

จากหลักการมุมมองแบบไฮเปอร์คิวบ์ (Hypercube View) ซึ่งสามารถที่จะเปลี่ยนมุมมองของข้อมูล $\mathcal{Y}$ ให้อยู่ในรูปของเวกเตอร์ $\boldsymbol{y} \in \{0,1\}^K$ โดยที่องค์ประกอบที่ k จะมีค่าเป็น 1 ถ้า $k \in \mathcal{Y}$ จากภาพที่ 3.8 เราสามารถแสดงค่าของแต่ละ $\mathcal{Y}$ ในรูปแบบของ K มิติในมุมมองแบบไฮเปอร์คิวบ์ โดยที่องค์ประกอบที่ k ของเวกเตอร์ $\boldsymbol{y}$ จะตรงกับแกนในมุมมองไฮเปอร์คิวบ์ ซึ่งจะบอกว่ามี หรือ ไม่มีองค์ประกอบที่ k ใน $\mathcal{Y}$



4

ภาพที่ 3.8 แผนผังแสดงหลักการมุมมองแบบไฮเปอร์คิวบ์ (hypercube view)

ตามหลักการมุมมองแบบไฮเปอร์คิวบ์ (Hypercube View) คำตอบของชุดข้อมูลหลายคำตอบ จะมีความสัมพันธ์เชื่อมโยงกันในแต่ละคำตอบ ซึ่งสามารถเปลี่ยนมุมมองได้ วิธีที่จะนำมาใช้ในการเปลี่ยนมุมมองในงานวิจัยชิ้นนี้ เรียกว่า การแปลงมิติเชิงเส้น (Principle Label Space Transformation – PLST) (Farbound Tai, 2010) ซึ่งจะใช้เทคนิคซิงกูลาร์ แวลู ดีคอมโพสิชัน (Singular Value Decomposition – SVD) มาหาตัวดำเนินการแปลงมิติเชิงเส้น (Transformation Operator)

จากการนำชุดคำตอบของข้อมูลหลายคำตอบ (Extreme Multi-label) ขอแทนด้วยสัญลักษณ์ Y ซึ่งมีขนาด $N \times K$ และมีมิติเท่ากับ K มาแตกซิงกูลาร์ แวลู ออกเป็นองค์ประกอบ 3 ส่วน ดังสมการ 3.1

$$Y = U\Sigma V^T \tag{3.1}$$

โดยที่ U มีขนาดเป็น $N \times N$ และ V^T มีขนาดเป็น $K \times K$ เป็นเมทริกซ์ที่มีคอลัมน์ตั้งฉากกัน (Orthonormal) ส่วน Σ มีขนาดเป็น $N \times K$ เป็นเมทริกซ์แนวทแยง ดังได้กล่าวไว้ในบทที่ 2

ตามหลักการของซิงกูลาร์ แวลู ดีคอมโพสิชัน แต่ละแถว y_n สามารถแสดงให้อยู่ในรูปแบบองค์ประกอบเชิงเส้นของแต่ละคอลัมน์ v_k^T เมทริกซ์แนวทแยง Σ ประกอบด้วย ค่าซิงกูลาร์แวลู σ_k ตามแนวทแยงโดยที่ $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_k$ หากเขียนสมการ 3.1 ใหม่เป็น

$$YV = U\Sigma \quad 3.2$$

เมทริกซ์ Vสามารถมองเป็นตัวดำเนินการแปลงเมทริกซ์ Y ให้ไปอยู่ในมิติอื่นได้ เนื่องจากค่าซิงกูลาร์ แวลู K ตัว จะสัมพันธ์กับแกนหลักของมิติตั้งต้นทุกมิติ โดยค่าซิงกูลาร์ แวลู ของแกนหลักที่ 1ที่มีความสำคัญมากที่สุด จะมีค่ามากที่สุด และ ค่าซิงกูลาร์ แวลู ของแกนหลักถัด ๆ ไป จะมีค่าน้อยลงตามลำดับความสำคัญ หากต้องการลดจำนวนมิติของเมทริกซ์ Y ให้เหลือ M มิติ เราสามารถเลือกตัดแกนหลักที่สัมพันธ์กับค่าค่าซิงกูลาร์ แวลู ที่ถัดจากตัวที่ M เพื่อเลือกแกนหลักที่มีค่าซิงกูล่าร์ แวลู สูงที่สุด M ตัวแรก นำมาสร้างตัวดำเนินการแปลงข้อมูล (Projection matrix) ขอแทนค่าด้วย P และตัวดำเนินการแปลงข้อมูลกลับ (Decoder matrix) ขอแทนค่าด้วย D มาใช้ลดมิติของ Y ให้เหลือ M มิติ จึงได้ว่า

$$P = V_M = [v_1\ v_2\ \dots\ v_m\ ] \text{ และ } D = V_M^T = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_m \end{bmatrix} \quad 3.3$$

3.4 โครงสร้างของแบบจำลอง

งานวิจัยนี้ได้ประยุกต์ใช้โครงสร้างแบบจำลองการเรียนรู้เชิงลึกจากงานวิจัย Cold Start Thread Recommendation as Extreme Multi-label Classification (Kishaloy Halder, 2018) ซึ่งได้ปรับเปลี่ยนโครงสร้างให้เหมาะกับงานวิจัย โดยการควบรวมวิธีการจำแนกประเภทข้อมูลหลายคำตอบ และวิธีลดมิติข้อมูลคำตอบ (Combining of eXtreme Multi-Label Classification and Principle lAbel space transfOrmation - XMLC-PAO) และขอเรียกวิธีการนี้สั้น ๆ ว่า “XMLC-PAO” โครงสร้างของแบบจำลอง แบ่งออกเป็นสองแบบ เพื่อเปรียบเทียบประสิทธิภาพการทำงาน รายละเอียดโครงสร้างของแบบจำลองทั้งสองแบบได้แสดงไว้ในภาพที่ 3.10 และโครงสร้างของแบบจำลองตั้งต้นได้แสดงไว้ในภาพที่ 3.9 (จำนวนพารามิเตอร์ของแต่ละแบบจำลองสามารถตรวจสอบได้จากภาคผนวกของรายงาน)

3.4.1 แบบจำลองตั้งต้น (Original Model)

ก่อนจะกล่าวถึงแบบจำลองที่ได้ปรับเปลี่ยนโครงสร้างสำหรับงานวิจัยนี้ ขอกล่าวถึงแบบจำลองตั้งต้นจากงานวิจัย Cold Start Thread Recommendation as Extreme Multi-label Classification (Kishaloy Halder, 2018) ขั้นตอนการทำงานของแบบจำลองตั้งต้น แบบออกได้เป็น 4 ส่วนด้วยกัน ซึ่งแสดงไว้ในแผนผังในภาพที่ 3.9

(1) การแทนค่าคำ (Word Embedding)

เป็นขั้นตอนเริ่มต้น ของการสร้างคุณลักษณะข้อมูลของแบบจำลองการเรียนรู้เชิงลึก โดยการแทนค่าคำจะใช้คลังข้อความ (Corpus) แบบโกลว (GloVe) เพื่อจะแปลงข้อความให้อยู่ในรูปของเวกเตอร์

(2) การจับใจความสำคัญของประโยค โดยใช้โครงข่ายประสาทวนรอบสองทิศทางที่มีหน่วยความจำแบบย่อ (Bi-GRU)

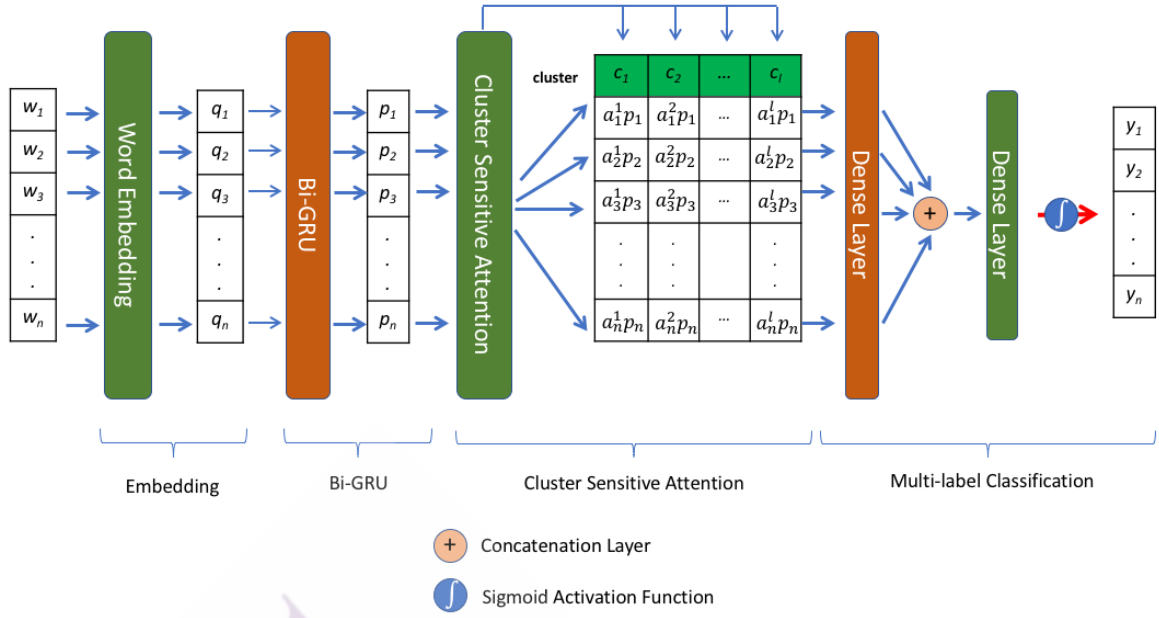
หลังจากแปลงข้อความให้อยู่ในรูปของเวกเตอร์แล้ว จึงทำการจับใจความของประโยค เพื่อให้คุณลักษณะของข้อความมีความถูกต้องมากขึ้น โดยใช้โครงข่ายประสาทวนรอบสองทิศทาง

(3) การจัดกลุ่มข้อความตามใจความสำคัญ (Cluster Sensitive Attention)

ขั้นตอนนี้เป็นการแบ่งข้อความออกเป็นกลุ่ม ๆ ตามสมมติฐานว่า ผู้ใช้ที่อยู่กลุ่มเดียวกัน จะสนใจตอบคำถามประเภทเดียวกัน โดยทำการแบ่งกลุ่มจากใจความของข้อความที่ได้มาจากขั้นตอนที่ *(2)*

(4) แบบจำลองการประเมินค่าของข้อมูลหลายคำตอบ (Extreme Multi-label Classification Model)

หลังจากได้คุณสมบัติที่เหมาะแล้วแล้ว จึงส่งข้อมูลจากขั้นตอนที่ *(3)* เข้าสู้โครงข่ายประสาทที่จะทำการเรียนรู้เพื่อสร้างแบบจำลองสำหรับทำนายคำตอบ ในส่วนนี้ใช้ฟังก์ชันกระตุ้นแบบ ซิกมอยด์ (Sigmoid Activation Function) และใช้ค่าการประเมินแบบจำลองการเรียนรู้ของแบบจำลองเป็น ค่าไบนารี ครอสเอนโทรปี (Binary Cross-Entropy) เพื่อให้คำตอบที่ออกมา เป็นค่าความน่าจะเป็นที่ผู้ใช้แต่ละคนจะเข้ามาตอบกระทู้ โดยที่คำตอบของแบบจำลองตั้งต้น คือ ชุดข้อมูลหลายคำตอบ (Y) ที่ยังไม่ได้ลดขนาดลง



ภาพที่ 3.9 โครงสร้างของแบบจำลองตั้งต้น (Original Model) ในงานวิจัย Cold Start Thread Recommendation as Extreme Multi-label Classification (Kishaloy Halder, 2018) ซึ่งมีชุดคำตอบเป็น Y

หลังจากทำความรู้จักแบบจำลองตั้งต้น (Original Model) ไปแล้ว ต่อไปจะขอกล่าวถึง แบบจำลองที่ได้ปรับปรุงโครงสร้างให้เหมาะกับงานวิจัยนี้ ในหัวข้อที่ 3.4.2 และ 3.4.3

3.4.2 แบบจำลอง XMLC-PAO แบบที่ 1 (XMLC-PAO-1)

ขั้นตอนการทำงานของแบบจำลองแบบที่ 1 ในภาพที่ 3.10 (ก) สามารถแบ่งออกได้เป็น 6 ส่วนด้วยกัน โดยเป็นส่วนของการแปลงมิติชุดคำตอบ (Label Space Transformation) (ข้อ *(1)* และ *(6)*) และส่วนของแบบจำลองการเรียนรู้เชิงลึก (Deep Neural Network) (ข้อ *(2) – (5)*)

(1) การลดจำนวนมิติของชุดข้อมูลคำตอบ เพื่อแก้ปัญหาการมีมิติจำนวนมากของระบบแนะนำ วิธีการ “XMLC-PAO” เริ่มต้นจากการลดขนาดมิติของชุดคำตอบ Y ด้วยวิธีการที่ดัดแปลงมาจาก การแปลงมิติเชิงเส้น (PLST) ตามสมการที่ 3.4 เพื่อให้ได้ชุดคำตอบ H ซึ่งมีมิติที่ลดลง แล้วนำ H ไปใช้เป็นชุดคำตอบในส่วนท้ายสุดของโครงสร้างแบบจำลอง ในภาพที่ 3.10 (ก)

$$H = Y.V_M \qquad 3.4$$

เมื่อ V_M คือ ตัวดำเนินการแปลงมิติเชิงเส้นจาก Y เป็น H ในภาพที่ 3.10 (ก) ซึ่งที่มาของ V_M ได้แสดงไว้ในสมการที่ 3.3ตัวอย่างของการแปลงมิติเชิงเส้นระหว่าง Y และ H ได้แสดงไว้ในภาพที่ 3.11

(2) การแทนค่าคำ (Word Embedding)

เป็นขั้นตอนเริ่มต้น ของการสร้างคุณลักษณะของแบบจำลองการเรียนรู้เชิงลึก โดยการแทนค่าคำจะใช้คลังข้อความ (Corpus) แบบโกลว (GloVe) เพื่อจะแปลงข้อความให้อยู่ในรูปของเวกเตอร์

(3) การจับใจความสำคัญของประโยค โดยใช้โครงข่ายประสาทวนรอบสองทิศทางที่มีหน่วยความจำแบบย่อ (Bi-GRU)

หลังจากแปลงข้อความให้อยู่ในรูปของเวกเตอร์แล้ว จึงทำการจับใจความของประโยค เพื่อให้คุณลักษณะของข้อความมีความถูกต้องมากขึ้น โดยใช้โครงข่ายประสาทวนรอบสองทิศทาง

(4) การจัดกลุ่มข้อความตามใจความสำคัญ (Cluster Sensitive Attention)

ขั้นตอนนี้เป็นการแบ่งข้อความออกเป็นกลุ่ม ๆ ตามสมมติฐานว่า ผู้ใช้ที่อยู่กลุ่มเดียวกัน จะสนใจตอบคำถามประเภทเดียวกัน โดยทำการแบ่งกลุ่มจากใจความของข้อความที่ได้มาจากขั้นตอนที่ (3)

(5) แบบจำลองการประเมินค่าของข้อมูลหลายคำตอบ (Extreme Multi-label Regression Model)

หลังจากได้คุณสมบัติที่เหมาะแล้วแล้ว จึงส่งข้อมูลจากขั้นตอนที่ (4) เข้าสู่โครงข่ายประสาทที่จะทำการเรียนรู้เพื่อสร้างแบบจำลองสำหรับทำนายคำตอบ

เนื่องจากการแปลงมิติชุดคำตอบให้มีขนาดเล็กลง ทำให้ต้องดัดแปลงโครงสร้างให้อยู่ในรูปแบบของการประเมินค่า (Regression Model) เนื่องจากแต่ละองค์ประกอบของชุดคำตอบ H มีค่าที่ไม่อยู่ระหว่าง 0 ถึง 1 เช่นเดียวกับชุดคำตอบ Y ซึ่งเป็นชุดคำตอบที่เหมาะกับแบบจำลองการจำแนกประเภทข้อมูล (Classification Model) (ภาพที่ 3.11) แบบจำลองนี้จึงไม่มีการใช้ฟังก์ชันกระตุ้นใด ๆ และใช้ค่าการประเมินแบบจำลองการเรียนรู้ของแบบจำลอง เป็น ค่าความผิดพลาดเฉลี่ยกำลังสอง (Mean Square Error – MSE)

(6) การแปลงชุดข้อมูลคำตอบ H กลับไปอยู่ในมิติชุดคำตอบ Y

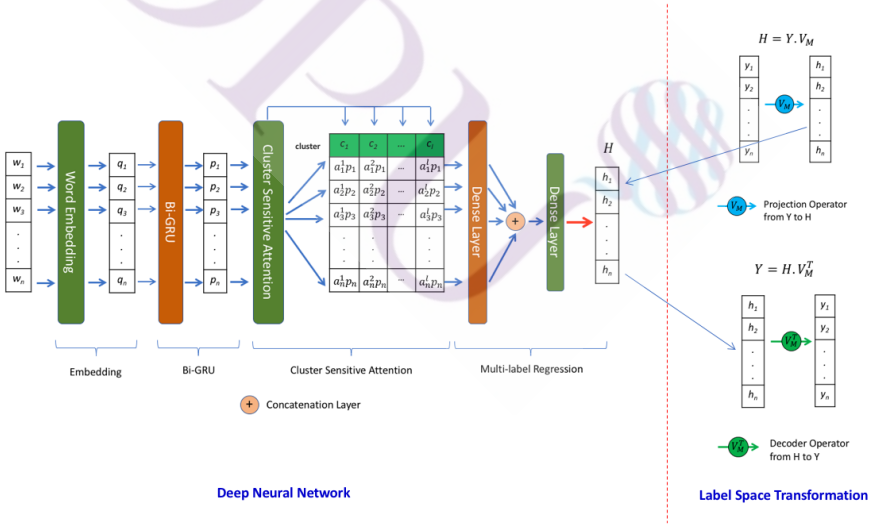
ในการประเมินประสิทธิภาพของแบบจำลอง จะต้องแปลงค่า H ที่แบบจำลองทำนายออกมาให้เป็นค่า Y โดยใช้สมการที่ 3.5 ก่อนที่จะนำไปประเมินประสิทธิภาพซึ่งจะได้กล่าวถึงรายละเอียดในหัวข้อ 3.5 ต่อไป

$$Y = H.V_M^T \qquad 3.5$$

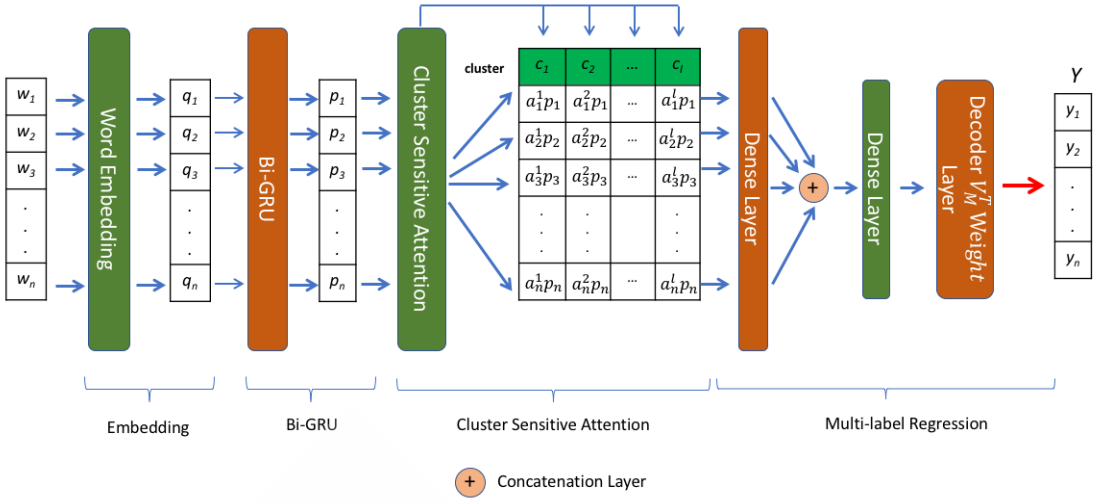
เมื่อ V_M^T คือ ตัวดำเนินการแปลงมิติเชิงเส้นจาก H เป็น Y ภาพที่ 3.10 (ก)

3.4.3 แบบจำลอง XMLC-PAO แบบที่ 2 (XMLC-PAO-2)

ขั้นตอนการทำงานของแบบจำลองแบบที่ 2 ในภาพที่ 3.10 (ข) แบ่งออกได้เป็น 4 ส่วน โดยตัดส่วนของการแปลงมิติชุดคำตอบ (Label Space Transformation) (ข้อ *(1)* และ *(6)*) ออกจากแบบจำลอง XMLC-PAO แบบที่ 1 แต่เติมโครงสร้างในชั้นสุดท้ายที่มีตัวแปลงชุดคำตอบ H เข้าไปเป็นค่าถ่วงน้ำหนัก (Weight) โดยกำหนดให้แบบจำลองไม่ทำการปรับเปลี่ยนค่าถ่วงน้ำหนักระหว่างกระบวนการเรียนรู้ของแบบจำลองเฉพาะในชั้นนี้ ชุดคำตอบของแบบจำลอง XMLC-PAO แบบที่ 2 จึงเป็น Y

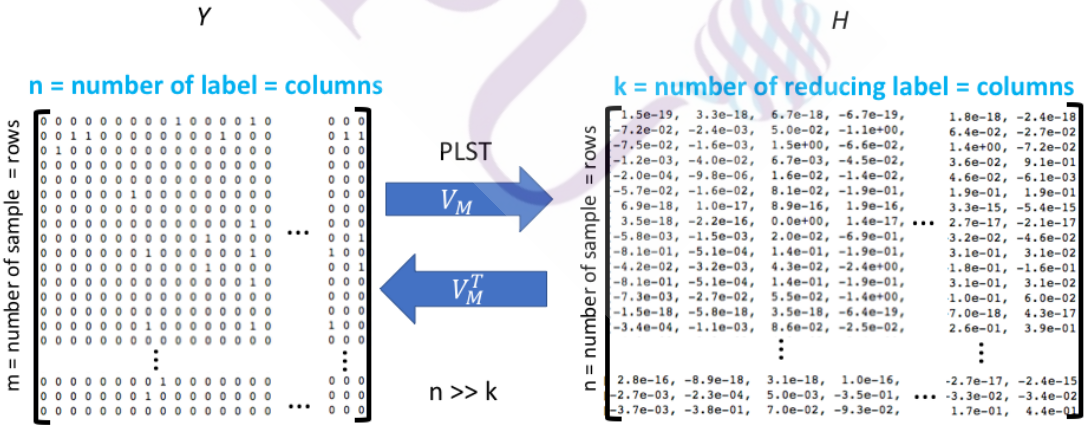


(ก) โครงสร้างของแบบจำลองแบบที่ 1 (XMLC-PAO-1) ตั้งแต่การเตรียมปัจจัยของข้อมูลขาเข้า จนถึงการเรียนรู้ของแบบจำลองที่มี H เป็นชุดข้อมูลคำตอบ



(ข) โครงสร้างของแบบจำลองแบบที่ 2 (XMLC-PAO-2) ตั้งแต่การเตรียมปัจจัยของข้อมูลขาเข้า จนถึงการเรียนรู้ของแบบจำลองที่มี Y เป็นชุดข้อมูลคำตอบ แบบจำลองนี้ได้รวมเอาตัวแปลงคำตอบ H เป็น $Y\ V_M^T$ เป็นของชั้นสุดท้ายของแบบจำลอง

ภาพที่ 3.10 แผนผังโครงสร้างแบบจำลองงานวิจัยด้วยวิธีการ “XMLC-PAO”



ภาพที่ 3.11 ความสัมพันธ์ของการแปลงมิติเชิงเส้นระหว่าง Y และ H

3.5 การวัดประสิทธิภาพของแบบจำลอง

การวัดประสิทธิภาพการทำงานของแบบจำลองในงานวิจัยนี้ ไม่สามารถใช้ค่าความถูกต้อง (Accuracy) และค่าความแม่นยำ (Precision) ได้ เนื่องจากชุดข้อมูลมีข้อมูลคำตอบแบบเบา

บาง (Sparse) การวัดประสิทธิภาพจึงใช้ปริมาณเหล่านี้เป็นตัววัดประสิทธิ์ภาพแทน คือ ค่าเฉลี่ยส่วนกลับของลำดับ (Mean Reciprocal Rank – MRR), รีคอลล์ที่จำนวน M ตัวแรก (Recall@M) และความน่าจะเป็นสะสมที่ลดทอนด้วยค่าปกติที่จำนวน M ตัวแรก (Normalized Discounted Cumulative Gain – NDCG@M)

เนื่องจากคำตอบที่ได้ออกมาจากระบบแนะนำ จะมีค่าเป็นความน่าจะเป็นที่ผู้ใช้ในระบบคนหนึ่งจะสนใจกระทู้หนึ่ง ค่าคำตอบที่ออกมา จะอยู่ระหว่าง 0-1 ซึ่งสามารถนำคำตอบจากระบบแนะนำมาเรียงลำดับตามความน่าจะเป็นที่ผู้ใช้จะสนใจกระทู้ ลำดับของคำตอบที่ได้จากระบบแนะนำนี้ จะนำไปใช้ในการวัดประสิทธิภาพของแบบจำลองโดยปริมาณที่ได้กล่าวมาข้างต้นต่อไป

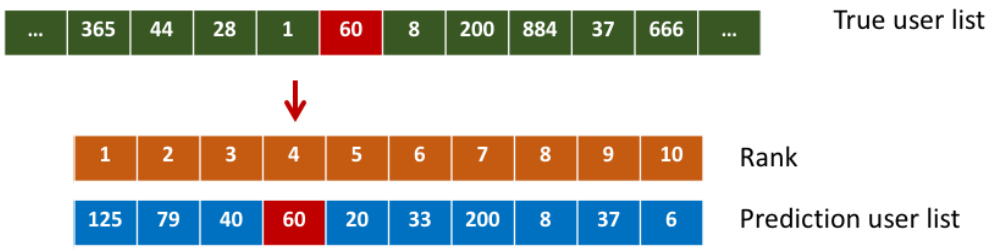
3.5.1. ค่าเฉลี่ยส่วนกลับของลำดับ (Mean Reciprocal Rank – MRR)

ปริมาณนี้ใช้บอกถึงความสามารถในการระบุผู้ใช้ที่สนใจในลำดับแรกของระบบแนะนำ โดยเริ่มพิจารณาจาก ค่าส่วนกลับของลำดับ สำหรับแต่ละคำตอบของแต่ละกระทู้ในระบบ ส่วนกลับของลำดับ คือ ส่วนกลับของปริมาณที่แสดงถึงตำแหน่งของผู้ใช้คนแรกที่ได้จากการทำนายเรียงลำดับตามคะแนนที่ตรงกับคำตอบที่ถูกต้อง สำหรับข้อมูลที่ใช้ในการทดลองที่มีแต่ละกระทู้เป็นข้อมูลเข้า และคำตอบเป็นรายการของผู้ใช้ ที่มีโอกาสเข้ามาตอบกระทู้ ส่วนกลับของลำดับของแต่ละกระทู้จึงมีค่าเดียว ดังตัวอย่างในภาพที่ 3.12 ลำดับแรกจากการทำนายของระบบแนะนำที่ผลการทำนายตรงกับคำตอบจริง (True user list) คือ ลำดับที่ 4 ส่วนกลับของลำดับ คือ 1/4

การหาตัวแทนของทั้งระบบ จำเป็นต้องคำนวณหาค่าเฉลี่ยส่วนกลับของลำดับ ของทุกกระทู้ที่อยู่ในระบบ ดังแสดงไว้ในสมการที่ 3.6

$$MRR = \frac{1}{n}\sum_{t=1}^{n}\frac{1}{r_t} \tag{3.6}$$

เมื่อ r_t คือลำดับที่สูงที่สุดที่คำตอบที่ได้จากการทำนายที่ตรงกับคำตอบจริง สำหรับกระทู้ที่ t ค่าเฉลี่ยส่วนกลับของลำดับ เป็นค่าเฉลี่ยของส่วนกลับของทุกกระทู้ที่อยู่ในชุดข้อมูลสำหรับเรียนรู้ หรือ ชุดข้อมูลทดสอบ



ภาพที่ 3.12 แผนผังแสดงหลักการคำนวณค่าส่วนกลับของลำดับ ของกระทู้ที่ t ในชุดข้อมูล ซึ่งมีค่าลำดับ r_t = 4 และ ส่วนกลับของลำดับ 1/r_t = 1/4

3.5.2. รีคอลล์ที่ M ลำดับแรก (Recall@M)

รีคอลล์ที่ M ลำดับแรก คือ จำนวนผู้ใช้ที่อยู่ในกลุ่ม M ลำดับแรกที่แบบจำลองทำนายว่ามีปฏิสัมพันธ์กับแต่ละกระทู้ในระบบที่ตรงกับคำตอบจริงของชุดข้อมูลสำหรับเรียนรู้ หรือชุดข้อมูลมูลทดสอบ ภาพที่ 3.13 แสดงแผนผังเพื่อความเข้าใจของปริมาณ รีคอลล์ที่ M ลำดับแรก

จากภาพที่ 3.13 สำหรับกลุ่ม 10 คนแรก (M = 10) จากลำดับ (Rank) ที่ระบบแนะนำทำนายว่ามีปฏิสัมพันธ์กับกระทู้หนึ่งในระบบ มีจำนวนคำตอบที่ตรงกับคำตอบที่ถูกต้อง (True user list) 4 คน ดังนั้น ค่า Recall@M ในภาพคือ 0.4 หรือ 4%

ภาพที่ 3.13 แผนผังแสดงหลักการคำนวณค่ารีคอลล์ที่จำนวน M ตัวแรก (Recall@M) เมื่อ M = 10 ของกระทู้ที่ t

ค่าเฉลี่ยของ Recall@M สำหรับทุกกระทู้ จะใช้เป็นตัวแทนของระบบ ค่ายิ่งสูงแสดงว่าประสิทธิภาพของแบบจำลองยิ่งดี

3.5.3. ความน่าจะเป็นสะสมที่ลดทอนด้วยค่าปกติที่ M ลำดับแรก (Normalized Discounted Cumulative Gain – NDCG@M)

เป็นการประเมินระบบการแนะนำ ซึ่งมีลักษณะคล้ายกับรีคอลล์ที่ M ลำดับแรก (Recall@M) ต่างกันตรงที่มีการให้น้ำหนักที่มากกว่าสำหรับลำดับผู้ใช้ที่สูงกว่าจากการทำนายของระบบแนะนำ ค่า NDCG@M สำหรับแต่ละกระทู้ t คำนวณได้จาก สมการ 3.7

$$NDCG_t = Z_t \sum_{j=1}^{M} \frac{2^{r(j)}-1}{\log(1+j)} \quad 3.7$$

เมื่อ $Z_t = 1/IDCG_t$ คือ ค่าคงที่ซึ่งบอกถึงส่วนกลับของค่าที่มากที่สุดที่เป็นไปได้ของความน่าจะเป็นสะสม (Ideal Discounted Cumulative Gain) สำหรับแต่ละกระทู้ t ทำให้กระทู้ที่ค่า$\sum_{j=1}^{M} \frac{2^{r(j)}-1}{\log(1+j)}$ เท่ากับ $IDCG_t$ เป็นกระทู้ที่มีลำดับที่สมบูรณ์ จึงมีค่า $NDCG_t$ เท่ากับ 1 และ สำหรับแต่ละ $r(j)$ จะมีค่าเป็นจำนวนเต็มที่บอกถึงความสัมพันธ์ โดยสำหรับงานวิจัยนี้ จะมีค่าเป็น 1 สำหรับกรณีที่มีความสัมพันธ์ และ 0 สำหรับกรณีไม่มีความสัมพันธ์ของผลการทำนายที่ลำดับ $j \in \{1, \dots, k\}$ *(https://en.wikipedia.org/wiki/Discounted_cumulative_gain)* โดยสำหรับผู้ใช้จากการทำนาย M คนแรก ค่า $NDCG_t$ คือค่าเฉลี่ย สำหรับทุกกระทู้ในชุดข้อมูลทดสอบ เพื่อให้ได้ค่า $NDCG@M$

การประเมินแบบจำลองในงานวิจัยนี้ จะทดสอบที่ค่า M = [5, 10, 30, 50, 100] เพื่อทดสอบคุณภาพของระบบการแนะนำที่ระดับเริ่มต้นที่แตกต่างกันของลำดับในรายการค่าการทำนาย โดยทั้งค่า Recall@M และ NDCG@M ในตารางบันทึกผลการทดลองในบทที่ 4 จะมีหน่วยเป็นเปอร์เซ็นต์ (%)

3.6 ปริมาณทางสถิติสำหรับระบุสถานะของชุดข้อมูลคำตอบในการจำแนกประเภทข้อมูลหลายคำตอบ

นอกจากการวัดประสิทธิภาพการทำงานของแบบจำลองแล้ว สิ่งที่ต้องทำการพิจารณาคือคุณสมบัติของชุดข้อมูลคำตอบ ปริมาณที่ใช้ในการพิจารณาคือ สัดส่วนสมาชิกของชุดข้อมูลคำตอบ (Label Cardinality) และความหนาแน่นของชุดข้อมูลคำตอบ (Label Density) คุณสมบัติที่ต่างกันของชุดข้อมูลสามารถส่งผลกระทบต่อกระบวนการเรียนรู้สำหรับแบบจำลองของชุดข้อมูลหลายคำตอบได้ เช่น สัดส่วนสมาชิกของชุดข้อมูลที่มีค่าใกล้เคียงกัน แต่มีความหนาแน่นของข้อมูลที่ต่างกันมาก จะแสดงคุณลักษณะที่ต่างกัน (Bernardini, F. 2009)

3.6.1. สัดส่วนสมาชิกของชุดข้อมูลคำตอบ (Label Cardinality)

สัดส่วนสมาชิกของชุดข้อมูลคำตอบ (Label Cardinality) คือ ค่าเฉลี่ยของจำนวนคำตอบสำหรับแต่ละตัวอย่างสำหรับข้อมูลชุดนั้น ซึ่งสามารถคำนวณได้จาก สมการที่ 3.8

$$Label\ Cardinality = \frac{1}{N}\sum_{i=1}^{N}|Y_i| \quad 3.8$$

3.6.2. ความหนาแน่นของชุดข้อมูลคำตอบ (Label Density)

ความหนาแน่นของชุดข้อมูลคำตอบ (Label Density) คือ ค่าเฉลี่ยของจำนวนคำตอบของแต่ละตัวอย่างสำหรับข้อมูลแต่ละชุด หารด้วยจำนวนคำตอบในชุดข้อมูลนั้น ซึ่งสามารถคำนวณได้จาก สมการที่ 3.9

$$Label\ Density = \frac{1}{N}\sum_{i=1}^{N}\frac{|Y_i|}{L} \quad 3.9$$

เมื่อ $L = \bigcup_{i=1}^{N} Y_i$

3.7 แนวทางการวิจัย

จากโครงสร้างของแบบจำลอง 3 แบบ (หัวข้อที่ 3.4) และการเตรียมชุดข้อมูล (หัวข้อที่ 3.2) ที่ได้กล่าวไปข้างต้น ขั้นตอนการทำงานวิจัยจะนำผลการทดลอง จากโครงสร้างของแบบจำลอง XMLC-PAO แบบที่ 1 และ XMLC-PAO แบบที่ 2 ไปเปรียบเทียบกับผลของแบบจำลองตั้งต้น (Original Model) ในงานวิจัย Cold Start Thread Recommendation as Extreme Multi-label Classification (Kishaloy Halder, 2018) ซึ่งมีขั้นตอนการศึกษาดังนี้

3.7.1. การสุ่มเลือกชุดข้อมูล

a. สุ่มข้อมูลจากชุดข้อมูลทั้งหมดในแบบต่าง ๆ กันดังนี้

i. สุ่มข้อมูล 5000 ตัวอย่าง จากชุดข้อมูลทั้งหมด แบบเชิงเส้น (Linear Sampling) ดังรายละเอียดด้านล่าง

- เลือกข้อมูลตัวอย่างที่ 1-5000 (Linear 1-5000)
- เลือกข้อมูลตัวอย่างที่ 201-5200 (Linear 201-5200)
- เลือกข้อมูลตัวอย่างที่ 801-5800 (Linear 801-5800)

- เลือกข้อมูลตัวอย่างที่ 5001-10000 (Linear 5001-10000)
- เลือกข้อมูลตัวอย่างที่ 10001-15000 (Linear 10001-15000)
- เลือกข้อมูลตัวอย่างที่ 12201-17200 (Linear 12201-17200)
- เลือกข้อมูลตัวอย่างที่ 15001-20000 (Linear 15001-20000)
- เลือกข้อมูลตัวอย่างที่ 20001-25000 (Linear 20001-25000)
- เลือกข้อมูลตัวอย่างที่ 25001-27932 (Linear 25001-27932)

จำนวนตัวอย่าง 2933 ตัวอย่าง

ii. สุ่มข้อมูล 5000 ตัวอย่าง จากชุดข้อมูลทั้งหมด แบบสุ่ม (Random Sampling) ดังรายละเอียดด้านล่าง

- เลือกจากค่าการสุ่มที่ 42 (Random seed-42)
- เลือกจากค่าการสุ่มที่ 101 (Random seed-101)

iii. สุ่มข้อมูล 6000 ตัวอย่าง จากชุดข้อมูลทั้งหมด แบบเชิงเส้น (Linear Sampling) ดังรายละเอียดด้านล่าง

- เลือกข้อมูลตัวอย่างที่ 1-6000
- เลือกข้อมูลตัวอย่างที่ 6001-12000
- เลือกข้อมูลตัวอย่างที่ 12001-18000
- เลือกข้อมูลตัวอย่างที่ 18001-24000
- เลือกข้อมูลตัวอย่างที่ 24001-27932 (จำนวนตัวอย่าง 3933 ตัวอย่าง)

iv. สุ่มข้อมูล 7000 ตัวอย่าง จากชุดข้อมูลทั้งหมด แบบเชิงเส้น (Linear Sampling) ดังรายละเอียดด้านล่าง

- เลือกข้อมูลตัวอย่างที่ 1-7000

v. ใช้ข้อมูลทั้งหมด 27932 ตัวอย่าง

ตารางที่ 3.1 แสดงรายละเอียด และคุณสมบัติของแต่ละชุดข้อมูลคำตอบ

Random Method	Number of Samples	Training Samples	Testing Samples	Number of Labels	Samples/ Labels	Label Cardinality	Label Density
Linear 1-5000	5000	4000	1000	6651	0.6014	4.4873	6.7467E-04
Linear 201-5200	5000	4000	1000	6646	0.6019	4.3305	6.5159E-04
Linear 801-5800	5000	4000	1000	6682	0.5986	3.9830	5.9608E-04
Linear 5001-10000	5000	4000	1000	6937	0.5766	3.1175	4.4940E-04
Linear 10001-15000	5000	4000	1000	7079	0.5651	2.7965	3.9504E-04
Linear 12201-17200	5000	4000	1000	7088	0.5643	2.6660	3.7613E-04
Linear 15001-20000	5000	4000	1000	7228	0.5534	2.6535	3.6711E-04
Linear 20001-25000	5000	4000	1000	6931	0.5771	2.4700	3.5637E-04
Linear 25001-27932	2933	2346	587	4792	0.4896	2.4224	5.0551E-04
Random seed-42	5000	4000	1000	8468	0.4724	3.1473	3.7166E-04
Random seed-101	5000	4000	1000	8422	0.4749	2.9568	3.5107E-04
Linear 1-6000	6000	4800	1200	7560	0.6349	4.2481	5.6192E-04
Linear 6001-12000	6000	4800	1200	7914	0.6065	3.0110	3.8047E-04
Linear 12001-18000	6000	4800	1200	8109	0.5919	2.6469	3.2641E-04
Linear 18001-24000	6000	4800	1200	8116	0.5914	2.4935	3.0724E-04
Linear 24001-27932	3933	3146	787	5983	0.5258	2.4619	4.1148E-04
Linear 1-7000	7000	5600	1400	8465	0.6615	4.1248	4.8728E-04
Linear 1-27932	27932	22345	5587	23335	0.9576	3.0060	1.2882E-04

ตารางที่ 3.2 แสดงจำนวนคำเฉลี่ยในกระทู้ สำหรับชุดเรียนรู้ และชุดทดสอบ ของแต่ละชุดข้อมูล

Random Method	Training Samples Avg. Words	Testing Samples Avg. Words
Linear 1-5000	132	134
Linear 201-5200	135	130
Linear 801-5800	138	134
Linear 5001-10000	150	143
Linear 10001-15000	159	155
Linear 12201-17200	159	163
Linear 15001-20000	163	172
Linear 20001-25000	164	168
Linear 25001-27932	168	168
Random seed-42	156	156
Random seed-101	159	153
Linear 1-6000	134	135
Linear 6001-12000	151	156
Linear 12001-18000	161	156
Linear 18001-24000	166	165
Linear 24001-27932	168	169
Linear 1-7000	137	133
Linear 1-27932	155	156

3.7.2. แบ่งชุดข้อมูลออกเป็นชุดข้อมูลสำหรับเรียนรู้ และชุดข้อมูลสำหรับทดสอบ

จากชุดข้อมูลที่เลือกไว้ในหัวข้อ 3.7.1 แต่ละชุด ทำการแบ่งออกเป็นชุดข้อมูลสำหรับการเรียนรู้ (Training Dataset) และชุดข้อมูลสำหรับการทดสอบ (Testing Dataset) ด้วยวิธีการ สปลิท แวลิเดชัน (Split Validation) ในสัดส่วน 80:20 คือ มีชุดข้อมูลเรียนรู้ 80% และชุดข้อมูลทดสอบ 20% โดยเลือกค่าการสุ่มตั้งต้นที่ 42 (random state = 42) รายละเอียดของแต่ละชุดข้อมูลได้แสดงไว้ในตารางที่ 3.1 และตารางที่ 3.2

3.7.3. คำนวณค่าสัดส่วนสมาชิกของชุดข้อมูลคำตอบ (Label Cardinality) และ ความหนาแน่นของชุดข้อมูลคำตอบ (Label Density)

รายละเอียดของแต่ละชุดข้อมูล และความแตกต่างของคุณสมบัติทางสถิติในรูปของ ค่าสัดส่วนสมาชิกของชุดข้อมูลคำตอบ (Label Cardinality) และ ความหนานานของชุดข้อมูลคำตอบ (Label Density) ได้แสดงเอาไว้ในตารางที่ 3.1

3.7.4. การเรียนรู้ของแบบจำลองตั้งต้น (Original Model)

สร้างแบบจำลองแบบเดียวกับในงานวิจัย Cold Start Thread Recommendation as Extreme Multi-label Classification (Kishaloy Halder, 2018) โดยมีรายละเอียดของกระบวนการเรียนรู้ดังนี้

a. สร้างแบบจำลองที่มีลักษณะดังแผนผังในภาพที่ 3.9 *(จำนวนพารามิเตอร์สามารถตรวจสอบได้จากภาคผนวก)*

b. กำหนดกระบวนการเรียนรู้ให้แบบจำลองโดยใช้วิธีการเรียนรู้แบบอดัม (Adam) ที่มีอัตราการเรียนรู้ (Learning Rate) อยู่ที่ 0.003

c. คำนวณค่าเฉลี่ยส่วนกลับของลำดับ (MRR), รีคอลล์ที่ M ลำดับแรก (Recall@M) และ ความน่าจะเป็นสะสมที่ลดทอนด้วยค่าปกติ (NDCG@M) สำหรับ M = [5, 10, 30, 50, 100] สำหรับข้อมูลชุดเรียนรู้ และข้อมูลชุดทดสอบ

3.7.5. การเรียนรู้ของแบบจำลอง XMLC-PAO แบบที่ 1 (XMLC-PAO-1)

สร้างแบบ XMLC-PAO แบบที่ 1 โดยมีรายละเอียดของกระบวนการเรียนรู้ดังนี้

a. สร้างแบบจำลองที่มีลักษณะดังแผนผังในภาพที่ 3.10 (ก) *(จำนวนพารามิเตอร์สามารถตรวจสอบได้จากภาคผนวก)* ซึ่งจะมีสัดส่วนของการลดขนาดมิติของข้อมูลคำตอบลง 3 ค่า ดังนี้

i. สัดส่วนมิติคงเหลือ 0.5 (Ratio 0.5) คือ ทำการแปลงมิติจาก Y เป็น H และลดจำนวนมิติข้อมูลคำตอบลงเหลือ 50%

ii. สัดส่วนมิติคงเหลือ 0.8 (Ratio 0.8) คือ ทำการแปลงมิติจาก Y เป็น H และลดจำนวนมิติข้อมูลคำตอบลงเหลือ 80%

iii. สัดส่วนมิติคงเหลือ 1.0 (Ratio 1.0) คือ ทำการแปลงมิติจาก Y เป็น H แต่ไม่มีการลดจำนวนมิติลง

b. สำหรับแต่ละสัดส่วนมิติคงเหลือของแบบจำลอง XMLC-PAO แบบที่ 1 แบบจำลอง 6 แบบจำลองจะถูกสร้างขึ้นโดยใช้วิธีการเรียนรู้แบบอดัม (Adam) ที่แต่ละแบบจำลอง มีอัตราการเรียนรู้ (Learning Rate) ต่างกัน คือ 0.0015, 0.003, 0.006, 0.008, 0.01 และ 0.02

c. คำนวณค่า ค่าเฉลี่ยส่วนกลับของลำดับ (MRR), รีคอลล์ที่ M ลำดับแรก (Recall@M) และ ความน่าจะเป็นสะสมที่ลดทอนด้วยค่าปกติ (NDCG@M) สำหรับ M = [5, 10, 30, 50, 100] สำหรับข้อมูลชุดเรียนรู้ และข้อมูลชุดทดสอบ สำหรับแบบจำลองทั้ง 18 แบบจำลอง

3.7.6. การเรียนรู้ของแบบจำลอง XMLC-PAO แบบที่ 2 (XMLC-PAO-2)

สร้างแบบ XMLC-PAO แบบที่ 2 โดยมีรายละเอียดของกระบวนการเรียนรู้ดังนี้

a. สร้างแบบจำลองที่มีลักษณะดังแผนผังในภาพที่ 3.10 (ข) *(จำนวนพารามิเตอร์สามารถตรวจสอบได้จากภาคผนวก)* ซึ่งจะมีสัดส่วนของการลดขนาดมิติของข้อมูลคำตอบลง 3 ค่า ดังนี้

i. สัดส่วนมิติคงเหลือ 0.5 (Ratio 0.5) คือ ทำการแปลงมิติจาก Y เป็น H และลดจำนวนมิติข้อมูลคำตอบลงเหลือ 50%

ii. สัดส่วนมิติคงเหลือ 0.8 (Ratio 0.8) คือ ทำการแปลงมิติจาก Y เป็น H และลดจำนวนมิติข้อมูลคำตอบลงเหลือ 80%

iii. สัดส่วนมิติคงเหลือ 1.0 (Ratio 1.0) คือ ทำการแปลงมิติจาก Y เป็น H แต่ไม่มีการลดจำนวนมิติลง

b. สำหรับแต่ละสัดส่วนมิติคงเหลือของแบบจำลอง XMLC-PAO แบบที่ 2 แบบจำลอง 6 แบบจำลองจะถูกสร้างขึ้น โดยใช้วิธีการเรียนรู้แบบอดัม (Adam) ที่แต่ละแบบจำลอง มีอัตราการเรียนรู้ (Learning Rate) ต่างกัน คือ 0.0015, 0.003, 0.006, 0.008, 0.01 และ 0.02

c. คำนวณค่า ค่าเฉลี่ยส่วนกลับของลำดับ (MRR), รีคอลล์ที่ M ลำดับแรก (Recall@M) และ ความน่าจะเป็นสะสมที่ลดทอนด้วยค่าปกติ (NDCG@M) สำหรับ M = [5, 10, 30, 50, 100] สำหรับข้อมูลชุดเรียนรู้ และข้อมูลชุดทดสอบ สำหรับแบบจำลองทั้ง 18 แบบจำลอง

3.7.7. เปรียบเทียบผลการทดลองสำหรับแบบจำลอง XMLC-PAO กับผลจากแบบจำลองตั้งต้น

ทำการเปรียบเทียบค่าเฉลี่ยส่วนกลับของลำดับ (MRR), รีคอลล์ที่ M ลำดับแรก (Recall@M) และ ความน่าจะเป็นสะสมที่ลดทอนด้วยค่าปกติ (NDCG@M) ของแบบจำลอง XMLC-PAO แบบที่ 1 และ XMLC-PAO แบบที่ 2 สำหรับทุก ๆ สัดส่วนมิติคงเหลือ และ ทุก ๆ ค่าของอัตราการเรียนรู้ กับผลจากแบบจำลองตั้งต้น ซึ่งผลการทดลองจะได้แสดงไว้ในบทถัดไป

บทที่ 4
ผลการศึกษา

การศึกษาการลดจำนวนมิติข้อมูลหลายคำตอบ โดยประยุกต์ใช้แบบจำลองการเรียนรู้เชิงลึก (Deep Learning Model) กับการจำแนกประเภทข้อมูลหลายคำตอบ (Extreme Multi-label Classification) จากงานวิจัย Cold Start Thread Recommendation as Extreme Multi-label Classification (Kishaloy Halder, 2018) ร่วมกับการลดจำนวนมิติข้อมูลหลายคำตอบ โดยใช้การแปลงมิติเชิงเส้น (Principle Label Space Transformation – PLST) (Farbound Tai, 2010) จะขอแบ่งการวิเคราะห์ผลการศึกษา ออกเป็น 5 ส่วน ซึ่งมีรายการดังต่อไปนี้

4.1. ตรวจสอบสมมติฐาน โดยการสุ่มเลือกข้อมูล 5000 ตัว ทั้งการเลือกข้อมูลแบบเชิงเส้น และการเลือกข้อมูลแบบสุ่ม

4.2. ตรวจสอบสมมติฐาน โดยการเลือกข้อมูลแบบเชิงเส้น 6000 ตัว

4.3. ตรวจสอบสมมติฐาน โดยการเลือกข้อมูลแบบเชิงเส้น 7000 ตัวแรก

4.4. ตรวจสอบสมมติฐาน โดยใช้ข้อมูลทั้งหมด

4.5. เปรียบผลของแบบจำลองที่ดีที่สุด สำหรับแต่ละกลุ่มตัวอย่าง

โดยจำนวนพารามิเตอร์ของแต่ละแบบจำลองสามารถตรวจสอบได้จากภาคผนวก

4.1. ตรวจสอบสมมติฐาน โดยการสุ่มเลือกข้อมูล 5000 ตัว ทั้งการเลือกข้อมูลแบบเชิงเส้น และการเลือกข้อมูลแบบสุ่ม

ผลการศึกษาในส่วนนี้ มีจุดประสงค์เพื่อทดสอบสมมติฐานเริ่มต้นของโครงสร้างแบบจำลองที่งานวิจัยนี้ได้ปรับปรุง ว่าสามารถเพิ่มประสิทธิ์ภาพการทำงานได้ดีขึ้น โดยการเปรียบเทียบผลการศึกษาของแต่ละแบบจำลองสัดส่วนการลดมิติ กับ ผลจากการแบบจำลองตั้งต้น

สำหรับแต่ละแถวของตารางบันทึกผลการทดลอง จะมีตัวเลขที่เน้นสีต่างกัน ตัวเลขที่เน้นด้วยสีแดง คือค่าที่ดีที่สุดในแต่ละแถว และตัวเลขที่เน้นด้วยสีน้ำเงิน คือค่าตัวเลขที่ดีที่สุดรองลงมาจากตัวเลขสีแดงในแต่ละแถว

4.1.1. ผลการทดลองชุดข้อมูล Linear 1-5000

ผลการทดลองของข้อมูล Linear 1-5000 ชุดทดสอบ (ตาราง 4.1 - 4.3) แบบจำลอง XMLC-PAO-2 สำหรับสัดส่วนการลดมิติ 0.8 และใช้อัตราการเรียนรู้เท่ากับ 0.02 (XMLC-PAO-2, 0.8, 0.02) ให้ผลดีที่สุด (ตาราง 4.2) โดยที่เมื่อเปรียบเทียบกับแบบจำลองตั้งต้น ค่า mrr เพิ่มขึ้น 0.9%, ค่า recall@10 – recall@100 เพิ่มขึ้น 0.2% - 32.0% และค่า ndcg@5-ndcg@100 เพิ่มขึ้น 1.2% - 14.5% ซึ่งจะเห็นได้ว่าประสิทธิภาพการทำงานของแบบจำลองเพิ่มขึ้นสำหรับค่า M ที่สูงขึ้น

ผลการทดลองของข้อมูลชุดเรียนรู้ สำหรับแบบจำลองเดียวกัน ให้ผลไปในทิศทางเดียวกัน และให้ผลที่ดีกว่าข้อมูลชุดทดสอบเล็กน้อย (ตาราง 4.56)

4.1.2. *ผลการทดลองชุดข้อมูล Linear 201-5200*

ผลการทดลองของข้อมูล Linear 201-5200 ชุดทดสอบ (ตาราง 4.4 - 4.6) แบบจำลอง XMLC-PAO-2 สำหรับสัดส่วนการลดมิติ 0.8 และใช้อัตราการเรียนรู้เท่ากับ 0.006 (XMLC-PAO-2, 0.8, 0.006) ให้ผลดีที่สุด (ตาราง 4.5) โดยที่เมื่อเปรียบเทียบกับแบบจำลองตั้งต้น ค่า mrr เพิ่มขึ้น 0.3%, ค่า recall@5 – recall@100 เพิ่มขึ้น 1.3% - 63.0% และค่า ndcg@5-ndcg@100 เพิ่มขึ้น 1.5% - 18.0% แต่ประสิทธิภาพไม่ได้เพิ่มขึ้นตามค่า M

ผลการทดลองของข้อมูลชุดเรียนรู้ ให้ผลที่แตกต่างจากข้อมูลชุดทดสอบ (ตาราง 4.58 – 4.60) แบบจำลอง XMLC-PAO-2 สำหรับสัดส่วนการลดมิติ 0.5 และใช้อัตราการเรียนรู้เท่ากับ 0.01 (XMLC-PAO-2, 0.5, 0.01) ให้ผลดีที่สุด (ตาราง 4.58)

4.1.3. *ผลการทดลองชุดข้อมูล Linear 801-5800*

ผลการทดลองของข้อมูล Linear 801-5800 ชุดทดสอบ (ตาราง 4.7 - 4.9)แบบจำลอง XMLC-PAO-2 สำหรับสัดส่วนการลดมิติ 0.8 และใช้อัตราการเรียนรู้เท่ากับ 0.008 (XMLC-PAO-2, 0.8, 0.008) ให้ผลดีที่สุด (ตาราง 4.8) โดยที่เมื่อเปรียบเทียบกับแบบจำลองตั้งต้น แม้ว่าปริมาณวัดผลจะไม่ได้สูงกว่าผลจากแบบจำลองตั้งต้นอย่างเห็นได้ชัดเจน

ผลการทดลองของข้อมูลชุดเรียนรู้ ให้ผลที่แตกต่างจากข้อมูลชุดทดสอบ (ตาราง 4.61 – 4.63) แบบจำลอง XMLC-PAO-2 สำหรับสัดส่วนการลดมิติ 0.8 และใช้อัตราการเรียนรู้เท่ากับ 0.01 (XMLC-PAO-2, 0.8, 0.01) ให้ผลดีที่สุด (ตาราง 4.62)

4.1.4. *ผลการทดลองชุดข้อมูล Linear 5001-10000*

ผลการทดลองของข้อมูล Linear 5001-10000 ชุดทดสอบ (ตาราง 4.10 - 4.12) แบบจำลอง XMLC-PAO-1 สำหรับสัดส่วนการลดมิติ 0.5 และใช้อัตราการเรียนรู้เท่ากับ 0.006 (XMLC-PAO-1, 0.5, 0.006) ให้ผลดีที่สุด (ตาราง 4.11) โดยที่เมื่อเปรียบเทียบกับแบบจำลองตั้งต้น

ค่า recall@M เพิ่มขึ้น 14.5% - 24.4% และค่า ndcg@M เพิ่มขึ้น 6.6% - 17.3% และประสิทธิภาพเพิ่มขึ้นตามค่า M แม้ว่า ค่า mrr, recall@10 และ ndcg@10 จะมีค่าน้อยกว่าผลจากแบบจำลองตั้งต้น ผลการทดลองของข้อมูลชุดเรียนรู้ ให้ผลที่แตกต่างจากข้อมูลชุดทดสอบ (ตาราง 4.64 – 4.66) แบบจำลอง XMLC-PAO-1 สำหรับสัดส่วนการลดมิติ 0.5 และใช้อัตราการเรียนรู้เท่ากับ 0.01 (XMLC-PAO-1, 0.5, 0.01) ให้ผลดีที่สุด (ตาราง 4.64)

4.1.5. ผลการทดลองชุดข้อมูล Linear 10001-15000

ผลการทดลองของข้อมูล Linear 10001-15000 ชุดทดสอบ (ตาราง 4.13 - 4.15) แบบจำลอง XMLC-PAO-2 สำหรับสัดส่วนการลดมิติ 1.0 และใช้อัตราการเรียนรู้เท่ากับ 0.006 (XMLC-PAO-2, 1.0, 0.006) ให้ผลดีที่สุด (ตาราง 4.14) โดยที่เมื่อเปรียบเทียบกับแบบจำลองตั้งต้น ค่า recall@10 – recall@100 เพิ่มขึ้น 1.2% - 5.1% และค่า ndcg@30-ndcg@50 เพิ่มขึ้น 0.1% - 0.4% แม้ว่า mrr, recall@M และ ndcg@M สำหรับ M ค่าอื่น จะมีค่าน้อยกว่าผลจากแบบจำลองตั้งต้น ผลการทดลองของข้อมูลชุดเรียนรู้ ให้ผลที่แตกต่างจากข้อมูลชุดทดสอบ (ตาราง 4.67 – 4.69) แบบจำลอง XMLC-PAO-2 สำหรับสัดส่วนการลดมิติ 0.8 และใช้อัตราการเรียนรู้เท่ากับ 0.008 (XMLC-PAO-2, 0.8, 0.008) ให้ผลดีที่สุด (ตาราง 4.68)

4.1.6. ผลการทดลองชุดข้อมูล Linear 12201-17200

ผลการทดลองของข้อมูล Linear 12201-17200 ชุดทดสอบ (ตาราง 4.16 - ตาราง 4.18) แบบจำลองตั้งต้นให้ผลที่ดีกว่าแบบจำลอง XMLC-PAO ทั้งสองแบบ
ผลการทดลองของข้อมูลชุดเรียนรู้ ให้ผลที่แตกต่างจากข้อมูลชุดทดสอบ (ตาราง 4.70 – 4.72) แบบจำลอง XMLC-PAO-2 สำหรับสัดส่วนการลดมิติ 0.8 และใช้อัตราการเรียนรู้เท่ากับ 0.006 (XMLC-PAO-2, 0.8, 0.006) ให้ผลดีที่สุด (ตาราง 4.71)

4.1.7. ผลการทดลองชุดข้อมูล Linear 15001-20000

ผลการทดลองของข้อมูล Linear 15001-20000 ชุดทดสอบ (ตาราง 4.19 - 4.21) แบบจำลองตั้งต้นให้ผลที่ดีกว่าแบบจำลอง XMLC-PAO ทั้งสองแบบ
ผลการทดลองของข้อมูลชุดเรียนรู้ (ตาราง 4.73 – 4.75) แบบจำลองตั้งต้นให้ผลที่ดีที่สุด

4.1.8. ผลการทดลองชุดข้อมูล Linear 20001-25000

ผลการทดลองของข้อมูล Linear 20001-25000 ชุดทดสอบ (ตาราง 4.22 - 4.24) แบบจำลองตั้งต้นให้ผลที่ดีกว่าแบบจำลอง XMLC-PAO ทั้งสองแบบ

ผลการทดลองของข้อมูลชุดเรียนรู้ (ตาราง 4.76 – 4.78) แบบจำลอง XMLC-PAO-1 สำหรับสัดส่วนการลดมิติ 0.8 และใช้อัตราการเรียนรู้เท่ากับ 0.006 (XMLC-PAO-1, 0.8, 0.006) ให้ผลดีที่สุด (ตาราง 4.77)

4.1.9. ผลการทดลองชุดข้อมูล Linear 25001-27932 (2933 ตัวอย่าง)

ผลการทดลองของข้อมูล Linear 25001-27932 ชุดทดสอบ ซึ่งมีข้อมูลเพียง *2933 ตัวอย่าง (ตาราง* 4.25 - 4.27) แบบจำลอง XMLC-PAO-2 สำหรับสัดส่วนการลดมิติ 0.5 และใช้อัตราการเรียนรู้เท่ากับ 0.006 (XMLC-PAO-2, 0.5, 0.006) ให้ผลดีที่สุด (ตาราง 4.25) โดยที่เมื่อเปรียบเทียบกับแบบจำลองตั้งต้น ค่า mrr เพิ่มขึ้น 9.7%, ค่า recall@5 – recall@50 เพิ่มขึ้น 0.2% - 9.6% และค่า ndcg@5-ndcg@50 เพิ่มขึ้น 7.1% - 17.8% แม้ว่า recall@100 และ ndcg@100 จะมีค่าน้อยกว่าผลจาแบบจำลองตั้งต้น

ผลการทดลองของข้อมูลชุดเรียนรู้ (ตาราง 4.79 – 4.81) แบบจำลอง XMLC-PAO-1 สำหรับสัดส่วนการลดมิติ 1.0 และใช้อัตราการเรียนรู้เท่ากับ 0.01 (XMLC-PAO-1, 1.0, 0.01) ให้ผลดีที่สุด (ตาราง 4.81)

4.1.10. ผลการทดลองชุดข้อมูล Random seed-42

ผลการทดลองของข้อมูล Random seed-42 ชุดทดสอบ (ตาราง 4.28 - 4.30) แบบจำลองตั้งต้นให้ผลที่ดีกว่าแบบจำลอง XMLC-PAO ทั้งสองแบบ

ผลการทดลองของข้อมูลชุดเรียนรู้ (ตาราง 4.82 – 4.84) แบบจำลองตั้งต้นให้ผลที่ดีที่สุด

4.1.11. ผลการทดลองชุดข้อมูล Random seed-101

ผลการทดลองของข้อมูล Random seed-101 ชุดทดสอบ (ตาราง 4.31 - 4.33) แบบจำลองตั้งต้นให้ผลที่ดีกว่าแบบจำลอง XMLC-PAO ทั้งสองแบบ

ผลการทดลองของข้อมูลชุดเรียนรู้ (ตาราง 4.85 – 4.87) แบบจำลองตั้งต้นให้ผลที่ดีที่สุด

4.2. ตรวจสอบสมมติฐาน โดยการเลือกข้อมูลแบบเชิงเส้น 6000 ตัว

ผลการศึกษาในส่วนนี้ มีจุดประสงค์เพื่อปรับเปลี่ยนจำนวนตัวอย่างในชุดข้อมูลเป็น 6000 ตัวอย่าง ว่าโครงสร้างแบบจำลองที่งานวิจัยนี้ได้ปรับปรุง สามารถเพิ่มประสิทธิ์ภาพการทำงานได้ดีขึ้นหรือไม่ โดยการเปรียบเทียบผลการศึกษาของแต่ละแบบจำลองสัดส่วนการลดมิติ กับผลจากการแบบจำลองตั้งต้น

สำหรับแต่ละแถวของตารางบันทึกผลการทดลอง จะมีตัวเลขที่เน้นสีต่างกัน ตัวเลขที่เน้นด้วยสีแดง คือค่าที่ดีที่สุดในแต่ละแถว และตัวเลขที่เน้นด้วยสีน้ำเงิน คือค่าตัวเลขที่ดีที่สุดรองลงมาจากตัวเลขสีแดงในแต่ละแถว

4.2.1. ผลการทดลองชุดข้อมูล Linear 1-6000

ผลการทดลองของข้อมูล Linear 1-6000 ชุดทดสอบ (ตาราง 4.34 - 4.36) แบบจำลองตั้งต้นให้ผลที่ดีกว่าแบบจำลอง XMLC-PAO ทั้งสองแบบ

ผลการทดลองของข้อมูลชุดเรียนรู้ (ตาราง 4.88 – 4.90) แบบจำลอง XMLC-PAO-1 สำหรับสัดส่วนการลดมิติ 1.0 และใช้อัตราการเรียนรู้เท่ากับ 0.0015 (XMLC-PAO-1, 1.0, 0.0015) ให้ผลดีที่สุด (ตาราง 4.90)

4.2.2. ผลการทดลองชุดข้อมูล Linear 6001-12000

ผลการทดลองของข้อมูล Linear 6001-12000 ชุดทดสอบ (ตาราง 4.37 - 4.39) แบบจำลองตั้งต้นให้ผลที่ดีกว่าแบบจำลอง XMLC-PAO ทั้งสองแบบ

ผลการทดลองของข้อมูลชุดเรียนรู้ (ตาราง 4.91 – 4.93) แบบจำลอง XMLC-PAO-1 สำหรับสัดส่วนการลดมิติ 0.5 และใช้อัตราการเรียนรู้เท่ากับ 0.0015 (XMLC-PAO-1, 0.5, 0.0015) ให้ผลดีที่สุด (ตาราง 4.91)

4.2.3. ผลการทดลองชุดข้อมูล Linear 12001-18000

ผลการทดลองของข้อมูล Linear 12001-18000 ชุดทดสอบ (ตาราง 4.40 - 4.42) แบบจำลองตั้งต้นให้ผลที่ดีกว่าแบบจำลอง XMLC-PAO ทั้งสองแบบ

ผลการทดลองของข้อมูลชุดเรียนรู้ (ตาราง 4.94 – 4.96) แบบจำลองตั้งต้นให้ผลที่ดีที่สุด

4.2.4. ผลการทดลองชุดข้อมูล Linear 18001-24000

ผลการทดลองของข้อมูล Linear 18001-24000 ชุดทดสอบ (ตาราง 4.43 - 4.45) แบบจำลอง XMLC-PAO-2 สำหรับสัดส่วนการลดมิติ 0.8 และใช้อัตราการเรียนรู้เท่ากับ 0.01 (XMLC-PAO-2, 0.8, 0.01) ให้ผลดีที่สุด (ตาราง 4.44) โดยที่เมื่อเปรียบเทียบกับแบบจำลองตั้งต้น ค่า mrr เพิ่มขึ้น 11.9%, ค่า recall@5 – recall@10 เพิ่มขึ้น 16.3% - 24.1% และค่า ndcg@5-ndcg@30 เพิ่มขึ้น 3.1% - 24.0% แม้ว่าค่า recall และ ndcg สำหรับ M ค่าอื่น ๆ ผลจากแบบจำลองตั้งต้นจะมีค่าสูงกว่าผลการทดลองของข้อมูลชุดเรียนรู้ (ตาราง 4.97 – 4.99) แบบจำลองตั้งต้นให้ผลที่ดีที่สุด

4.2.5. ผลการทดลองชุดข้อมูล Linear 24001-27932 (3933 ตัวอย่าง)

ผลการทดลองของข้อมูล Linear 24001-27932 ชุดทดสอบซึ่งมีจำนวนข้อมูลเพียง 3933 ตัวอย่าง (ตาราง 4.46 - 4.48) แบบจำลอง XMLC-PAO-1 สำหรับสัดส่วนการลดมิติ 0.5 และใช้อัตรา

การเรียนรู้เท่ากับ 0.01 (XMLC-PAO-1, 0.5, 0.01) ให้ผลดีที่สุด (ตาราง 4.46) โดยที่เมื่อเปรียบเทียบกับแบบจำลองตั้งต้น ค่า recall@5, 30, 50 เพิ่มขึ้น 42.5%, 13.5% และ 1.8% ตามลำดับ และค่า ndcg@5, 30, 50 เพิ่มขึ้น 26.9%, 6.5% และ 0.3% ตามลำดับ แม้ว่าค่า mrr, recall และ ndcg สำหรับ M ค่าอื่น ๆ ผลจากแบบจำลองตั้งต้นจะมีค่าสูงกว่า

ผลการทดลองของข้อมูลชุดเรียนรู้ (ตาราง 4.100 – 4.102) แบบจำลอง XMLC-PAO-2 สำหรับสัดส่วนการลดมิติ 0.5 และใช้อัตราการเรียนรู้เท่ากับ 0.003 (XMLC-PAO-2, 0.5, 0.003) ให้ผลดีที่สุด (ตาราง 4.100)

4.3. ตรวจสอบสมมติฐาน โดยการเลือกข้อมูล 7000 ตัวแรก

ผลการศึกษาในส่วนนี้ มีจุดประสงค์เพื่อปรับเปลี่ยนจำนวนตัวอย่างในชุดข้อมูลเป็น 7000 ตัวอย่าง โดยเลือกข้อมูล 7000 ตัวอย่างแรก ว่าโครงสร้างแบบจำลองที่งานวิจัยนี้ได้ปรับปรุง สามารถเพิ่มประสิทธิ์ภาพการทำงานได้ดีขึ้นหรือไม่ โดยการเปรียบเทียบผลการศึกษาของแต่ละแบบจำลองสัดส่วนการลดมิติ กับ ผลจากการแบบจำลองตั้งต้น

สำหรับแต่ละแถวของตารางบันทึกผลการทดลอง จะมีตัวเลขที่เน้นสีต่างกัน ตัวเลขที่เน้นด้วยสีแดง คือค่าที่ดีที่สุดในแต่ละแถว และตัวเลขที่เน้นด้วยสีน้ำเงิน คือค่าตัวเลขที่ดีที่สุดรองลงมาจากตัวเลขสีแดงในแต่ละแถว

ผลการทดลองชุดข้อมูลทดสอบ (ตาราง 4.49 – 4.51) แบบจำลอง XMLC-PAO-1 สำหรับสัดส่วนการลดมิติ 1.0 และใช้อัตราการเรียนรู้เท่ากับ 0.008 (XMLC-PAO-1, 1.0, 0.008) ให้ผลที่ใกล้เคียงกับแบบจำลองตั้งต้น โดยที่ ค่า recall@10, 30, และ 50 เพิ่มขึ้น 2.2%, 12.8% และ 0.4% ตามลำดับ และค่า ndcg@10 และ 30 เพิ่มขึ้น 2.6% และ 3.3% ตามลำดับ

ผลการทดลองของชุดข้อมูลเรียนรู้ (ตาราง 4.103 – 4.105) แบบจำลอง XMLC-PAO-2 สำหรับสัดส่วนการลดมิติ 1.0 และใช้อัตราการเรียนรู้เท่ากับ 0.0015 (XMLC-PAO-2, 1.0, 0.0015) ให้ผลดีที่สุด (ตาราง 4.105)

4.4. ตรวจสอบสมมติฐาน โดยใช้ข้อมูลทั้งหมด

ผลการศึกษาในส่วนนี้ มีจุดประสงค์เพื่อปรับเปลี่ยนจำนวนตัวอย่างเป็นใช้ข้อมูลทั้งหมดที่มี 27932 ตัวอย่าง ว่าโครงสร้างแบบจำลองที่งานวิจัยนี้ได้ปรับปรุง สามารถเพิ่มประสิทธิ์ภาพการทำงานได้ดีขึ้นหรือไม่ โดยการเปรียบเทียบผลการศึกษาของแต่ละแบบจำลองสัดส่วนการลดมิติ กับ ผลจากการแบบจำลองตั้งต้น

สำหรับแต่ละแถวของตารางบันทึกผลการทดลอง จะมีตัวเลขที่เน้นสีต่างกัน ตัวเลขที่เน้นด้วยสีแดง คือค่าที่ดีที่สุดในแต่ละแถว และตัวเลขที่เน้นด้วยสีน้ำเงิน คือค่าตัวเลขที่ดีที่สุดรองลงมาจากตัวเลขสีแดงในแต่ละแถว

ผลการทดลองของข้อมูลชุดทดสอบ (ตาราง 4.52 – 4.54) แบบจำลองตั้งต้นให้ผลที่ดีกว่าแบบจำลอง XMLC-PAO ทั้งสองแบบ เช่นเดียวกับผลการทดลองของข้อมูลชุดเรียนรู้ (ตาราง 4.106 – 4.108)

4.5. เปรียบเทียบผลชุดข้อมูลทดสอบของแบบจำลองที่ดีที่สุด สำหรับแต่ละกลุ่มตัวอย่าง

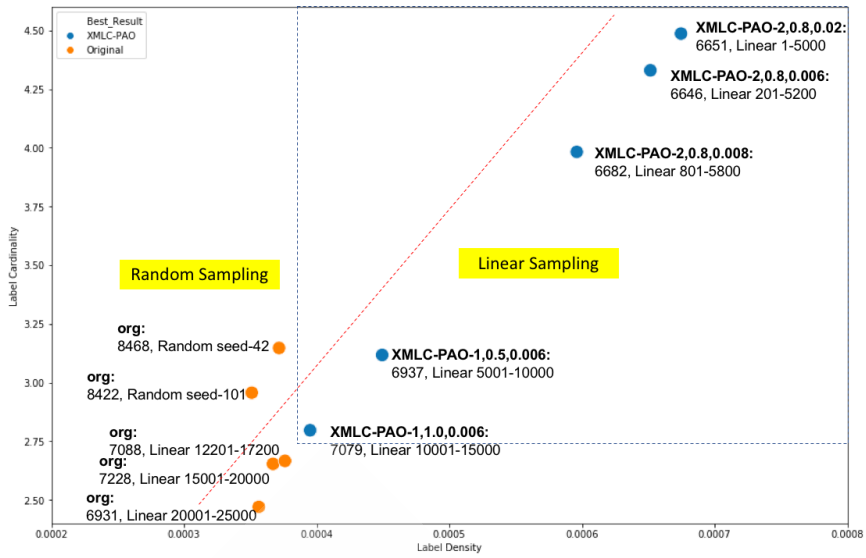
จากผลการทดลองชุดทดสอบทั้งหมด แบบจำลอง XMLC-PAO ทั้งสองแบบ จะทำงานได้ดีกว่า แบบจำลองตั้งต้น ในชุดข้อมูลที่มีจำนวนตัวอย่างไม่เกิน 6000 ตัวอย่าง และไม่ใช่ทุกชุดข้อมูลที่จำนวนตัวอย่างไม่เกิน 6000 จะให้ผลที่ดีกว่าแบบจำลองตั้งต้น ในหัวข้อนี้ เราจะลองเปลี่ยนมุมมองของการศึกษา เพื่อหาปัจจัยที่ทำให้ผลการทดลอง แตกต่างกันออกไป

เริ่มต้นจากเปรียบเทียบค่า สัดส่วนสมาชิกของชุดข้อมูลคำตอบ (Label Cardinality), ความหนาแน่นของชุดข้อมูลคำตอบ (Label Density) และ จำนวนมิติของชุดคำตอบ สำหรับแต่ละชุดข้อมูล ซึ่งแบ่งออกเป็น

4.5.1. เปรียบเทียบผลการทดลองชุดทดสอบของชุดข้อมูลที่มีจำนวนข้อมูล 5000 ตัว

4.5.2. เปรียบเทียบผลการทดลองชุดทอสอบของทุกชุดข้อมูล

4.5.1. เปรียบเทียบผลการทดลองชุดทดสอบของชุดข้อมูลที่มีจำนวนข้อมูล 5000 ตัว



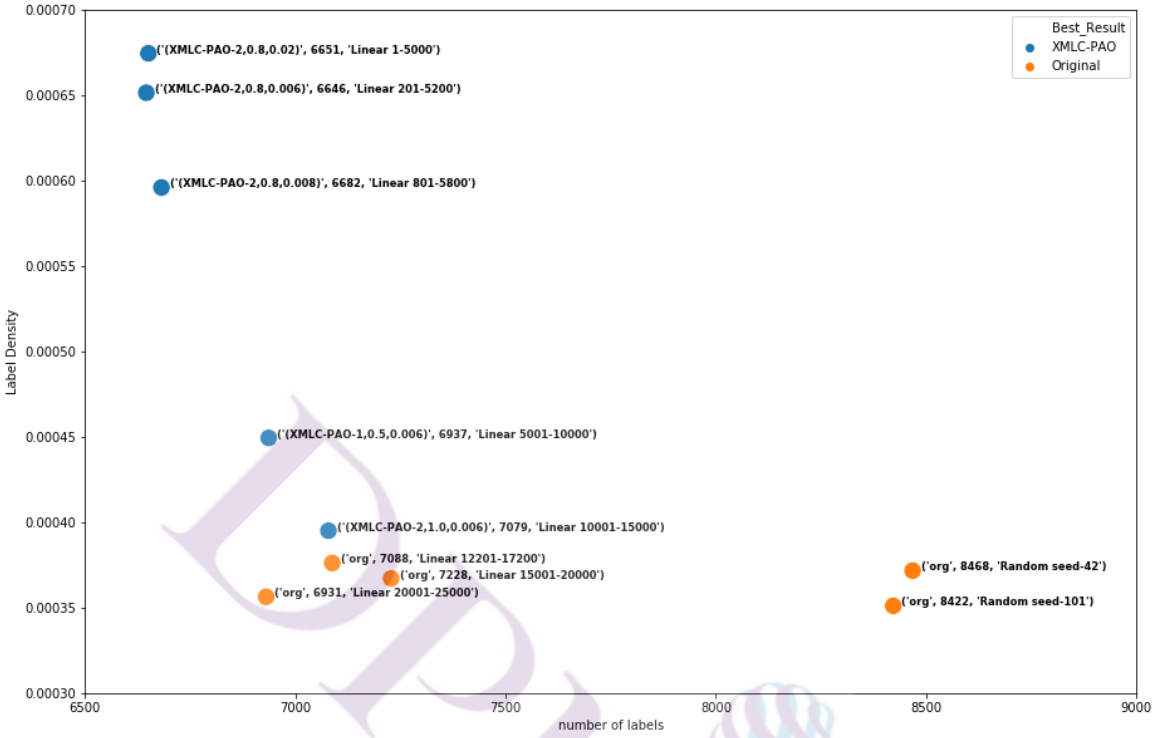
ภาพที่ 4.1 แสดงความสัมพันธ์ระหว่างสัดส่วนสมาชิกของชุดข้อมูลคำตอบ (Label Cardinality) และความหนาแน่นของชุดข้อมูลคำตอบ (Label Density) สำหรับแต่ละชุดข้อมูลที่มีจำนวนตัวอย่าง 5000 ตัว เพื่อเปรียบเทียบลักษณะข้อมูลที่ใช้ได้ดีกับแบบจำลอง XMLC-PAO และแบบจำลองตั้งต้น

จากภาพที่ 4.1 จุดสีน้ำเงิน แสดงความสัมพันธ์ของชุดข้อมูลที่แบบจำลอง XMLC-PAO ทำงานได้มีประสิทธิภาพดีกว่าแบบจำลองตั้งต้น ส่วนจุดสีส้ม แสดงความสัมพันธ์ของชุดข้อมูลที่แบบจำลองตั้งต้น ทำงานได้มีประสิทธิภาพดีกว่า ซึ่งสามารถมองเห็นการแยกตัวของกลุ่มชุดข้อมูลทั้งสองได้ชัดเจน

สำหรับชุดข้อมูลที่มีจำนวนข้อมูล 5000 ตัวอย่าง ชุดข้อมูลที่มีค่าความหนาแน่นของชุดข้อมูลคำตอบ (Label Density) มากกว่า 0.0004 และ มีค่าสัดส่วนสมาชิกของชุดข้อมูลคำตอบ (Label Cardinality) มากกว่า 2.75 แบบจำลอง XMLC-PAO ทำงานได้มีประสิทธิภาพดีว่าแบบจำลองตั้งต้น

นอกจากนี้ การเลือกข้อมูลแบบเชิงเส้น (Linear Sampling) และแบบสุ่ม (Random Sampling) สามารถส่งผลต่อการกระจายตัวของชุดข้อมูลที่แตกต่างกัน จากภาพที่ 4.1 การเลือกข้อมูลแบบเชิงเส้น (Linear Sampling) ส่งผลให้ความสัมพันธ์ระหว่างสัดส่วนสมาชิกของชุดข้อมูลคำตอบ (Label Cardinality) และความหนาแน่นของชุดข้อมูลคำตอบ (Label Density) สำหรับแต่ละ

ชุดข้อมูล มีแนวโน้มอยู่บนเส้นตรงเดียวกัน ในขณะที่การเลือกข้อมูลแบบสุ่ม (Random Sampling) ดูเหมือนจะแยกตัวออกจากกลุ่มกการเลือกแบบเชิงเส้น เส้นประสีแดงในภาพที่ 4.1 แสดงการแยกตัวของการเลือกข้อมูลที่แตกต่างกัน



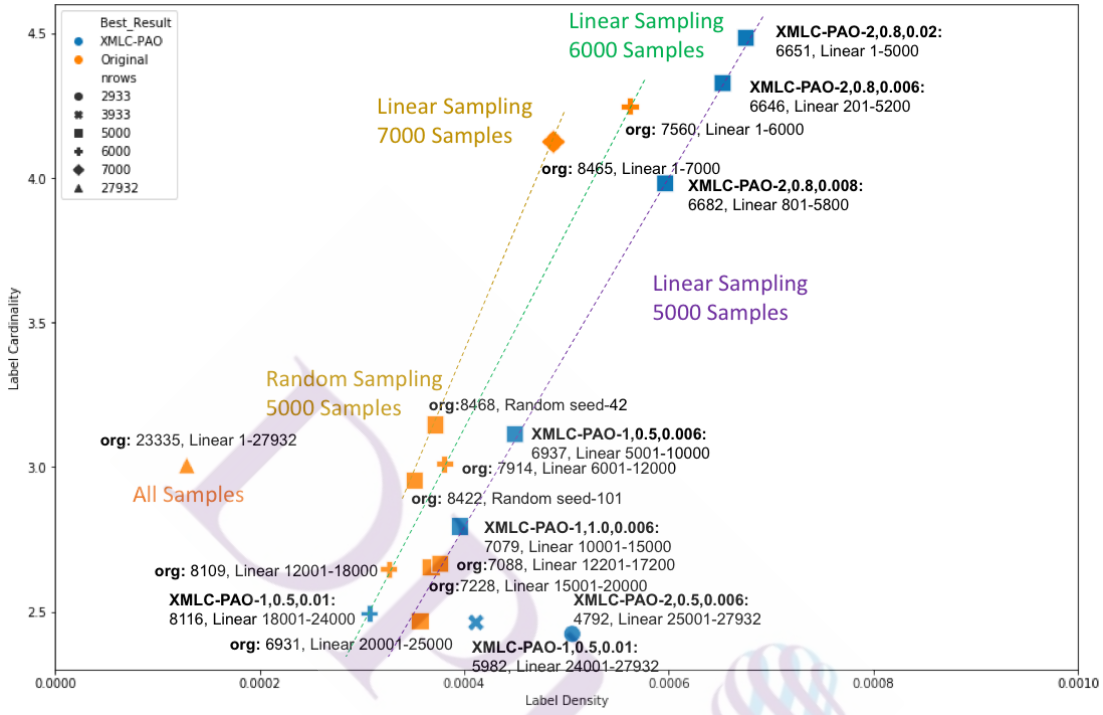
ภาพที่ 4.2 แสดงความสัมพันธ์ระหว่าง ความหนาแน่นของชุดข้อมูลคำตอบ (Label Density) และจำนวนคำตอบของชุดข้อมูลหลายคำตอบ (Number of Labels) สำหรับแต่ละชุดข้อมูลที่มีจำนวนตัวอย่าง 5000 ตัว

ภาพที่ 4.3 แสดงความสัมพันธ์ระหว่าง สัดส่วนสมาชิกของชุดข้อมูลคำตอบ (Label Cardinality) และ จำนวนคำตอบของชุดข้อมูลหลายคำตอบ (Number of Labels) สำหรับแต่ละชุดข้อมูลที่มีจำนวนตัวอย่าง 5000 ตัว

จากภาพที่ 4.2 และ 4.3 ซึ่งแสดงความสัมพันธ์ระหว่าง ความหนาแน่นของชุดข้อมูลคำตอบ (Label Density) และจำนวนคำตอบของชุดข้อมูลหลายคำตอบ (Number of Labels) และ ความสัมพันธ์ระหว่าง สัดส่วนสมาชิกของชุดข้อมูลคำตอบ (Label Cardinality) และ จำนวนคำตอบของชุดข้อมูลหลายคำตอบ (Number of Labels) ตามลำดับ จุดสีน้ำเงิน แสดงชุดข้อมูลที่แบบจำลอง XMLC-PAO ให้ประสิทธิภาพที่ดีกว่า และจุดสีส้ม แสดงชุดข้อมูลที่แบบจำลองตั้งต้นให้ประสิทธิภาพที่ดีกว่า ชุดข้อมูลทั้งสองกลุ่ม มีคุณลักษณะที่แยกตัวกันอย่างชัดเจน

4.5.2. เปรียบเทียบทุกชุดข้อมูล

เพื่อเปรียบเทียบภาพรวมของการศึกษา จึงได้นำผลการทดลองของทุกชุดข้อมูลมาเปรียบเทียบร่วมกัน และแสดงผลไว้ในภาพที่ 4.4-4.6



ภาพที่ 4.4 แสดงความสัมพันธ์ระหว่างสัดส่วนสมาชิกของชุดข้อมูลคำตอบ (Label Cardinality) และความหนาแน่นของชุดข้อมูลคำตอบ (Label Density) สำหรับแต่ละชุดข้อมูล เพื่อเปรียบเทียบลักษณะข้อมูลที่ใช้ได้ดีกับแบบจำลอง XMLC-PAO และแบบจำลองตั้งต้น

จากภาพที่ 4.4 จะเห็นได้ว่าชุดข้อมูลที่มีจำนวนข้อมูล 5000 ตัว และ 6000 ตัว และมีการเลือกข้อมูลแบบเชิงเส้น (Linear Sampling) มีการกระจายตัวของ ค่าสัดส่วนสมาชิกของชุดข้อมูลคำตอบ (Label Cardinality) และค่าความหนาแน่นของชุดข้อมูลคำตอบ (Label Density) อยู่บนเส้นตรงคนละเส้น นอกจากนี้ ชุดข้อมูลที่มีจำนวนข้อมูล 5000 ตัว และมีการเลือกข้อมูลแบบสุ่ม (Random Sampling) ดูเหมือนว่าจะมีการกระจายตัวของค่าสัดส่วนสมาชิกของชุดข้อมูลคำตอบ

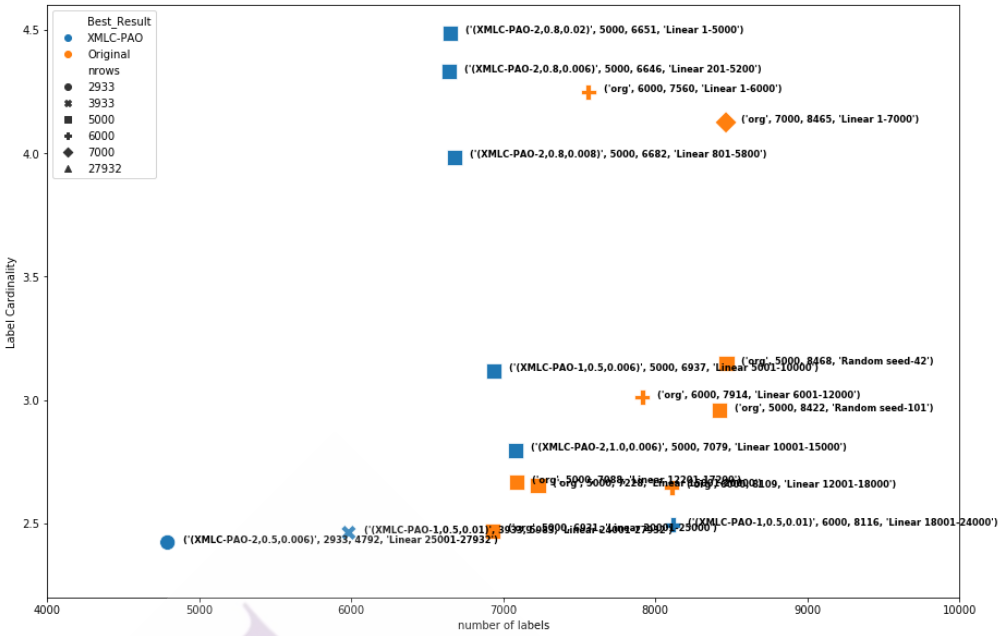
(Label Cardinality) และค่าความหนาแน่นของชุดข้อมูลคำตอบ (Label Density) อยู่บนเส้นตรงเดียวกันกับ ข้อมูลที่มีจำนวนข้อมูล 7000 ตัว

สำหรับชุดข้อมูลทั้งหมด 27932 ตัวอย่าง มีค่าความหนาแน่นของชุดข้อมูลคำตอบ (Label Density) ที่น้อยที่สุดเมื่อเทียบกับชุดข้อมูลอื่น ๆ การที่ความหนาแน่นน้อยกว่า อาจจะทำให้กระบวนการลดมิติของข้อมูลคำตอบมีความคลาดเคลื่อน ซึ่งเป็นผลให้วิธีการ XMLC-PAO ไม่สามารถเพิ่มประสิทธิภาพการทำงานจากแบบจำลองตั้งต้น

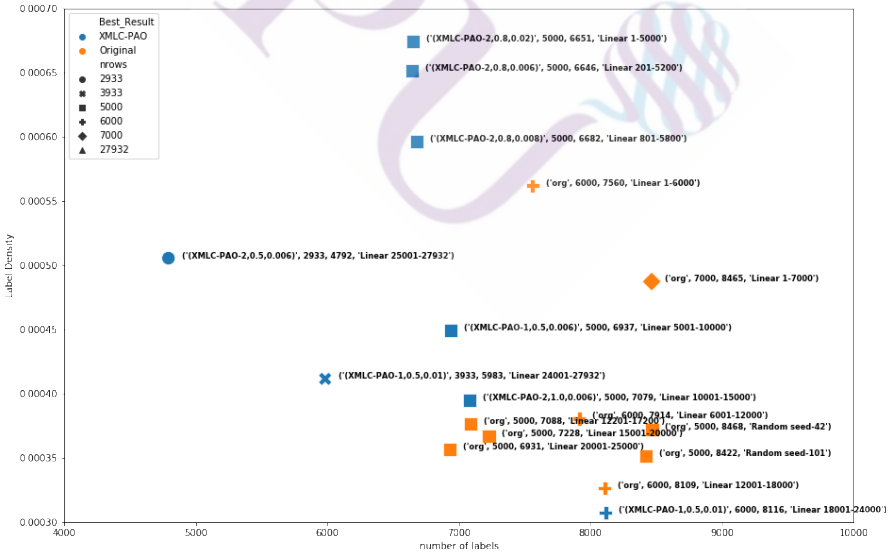
สำหรับชุดข้อมูลที่เลือกข้อมูลตัวอย่างที่ 25001-27932 (Linear 25001-27932) และ เลือกข้อมูลตัวอย่างที่ 24001-27932 (Linear 24001-27932) ซึ่งมีจำนวนข้อมูลน้อยกว่า 5000 ตัวอย่าง มีค่าความหนาแน่นของชุดข้อมูลคำตอบ (Label Density) ระหว่าง 0.0004-0.0005 แบบจำลอง XMLC-PAO-1 ที่ลดมิติเหลือ 50% มีประสิทธิภาพการทำงานที่ดีกว่าแบบจำลองตั้งต้น

สำหรับชุดข้อมูลที่เลือกข้อมูลแบบเชิงเส้น (Linear Sampling) ที่มีจำนวนข้อมูล 5000 ตัว และมีความหนาแน่นของชุดข้อมูลคำตอบมากกว่า 0.0005 แบบจำลอง XMLC-PAO-2 ที่ลดมิติเหลือ 80% มีประสิทธิภาพการทำงานที่ดีกว่าแบบจำลองตั้งต้น

จากภาพที่ 4.5 และ 4.6 ซึ่งแสดงความสัมพันธ์ระหว่าง ความหนาแน่นของชุดข้อมูลคำตอบ (Label Density) และจำนวนคำตอบของชุดข้อมูลหลายคำตอบ (Number of Labels) และความสัมพันธ์ระหว่าง สัดส่วนสมาชิกของชุดข้อมูลคำตอบ (Label Cardinality) และ จำนวนคำตอบของชุดข้อมูลหลายคำตอบ (Number of Labels) ตามลำดับ จุดสีน้ำเงิน แสดงชุดข้อมูลที่แบบจำลอง XMLC-PAO ให้ประสิทธิภาพที่ดีกว่า และจุดสีส้ม แสดงชุดข้อมูลที่แบบจำลองตั้งต้น ให้ประสิทธิภาพที่ดีกว่า ชุดข้อมูลทั้งสองกลุ่ม มีคุณลักษณะที่แยกตัวกันค่อยข้างชัดเจน



ภาพที่ 4.5 แสดงความสัมพันธ์ระหว่าง ความหนาแน่นของชุดข้อมูลคำตอบ(Label Density) และจำนวนคำตอบของชุดข้อมูลหลายคำตอบ (Number of Labels) สำหรับแต่ละชุดข้อมูล



ภาพที่ 4.6 แสดงความสัมพันธ์ระหว่าง สัดส่วนสมาชิกของชุดข้อมูลคำตอบ (Label Cardinality) และจำนวนคำตอบของชุดข้อมูลหลายคำตอบ (Number of Labels) สำหรับแต่ละชุดข้อมูล

ตารางผลการทดลองสำหรับข้อมูลชุดทดสอบ

ผลการทดลองข้อมูลชุดทดสอบ สำหรับข้อมูล 5000 ตัวอย่าง

ตาราง 4.1 : เปรียบเทียบผลข้อมูล Linear 1-5000 ชุดทดสอบ สำหรับสัดส่วนการลดมิติ 0.5 ระหว่างแบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับ แบบจำลองตั้งต้น

Random Method	Linear 1-5000	Testing Result											
XMLC-PAO Ratio	org	XMLC-PAO-1 : 0.5						XMLC-PAO-2 : 0.5					
Adam Learning	0.003	0.0015	0.003	0.006	0.008	0.01	0.02	0.0015	0.003	0.006	0.008	0.01	0.02
mrr	**0.0365**	0.0246	0.0265	0.0293	0.0303	0.0187	0.0265	0.0307	0.0268	0.0268	0.0268	0.0256	0.0245
recall@5	**2.0891**	1.2273	1.4648	1.5841	1.2633	0.8265	1.1411	1.7091	1.2262	1.4862	1.3883	1.0316	1.3563
recall@10	**3.0334**	1.5186	1.8984	2.2281	1.9994	1.2204	1.7546	2.1302	1.5388	1.8587	1.7367	1.2203	1.6706
recall@30	**5.4200**	2.2323	2.5025	4.6440	3.8904	2.0585	3.0559	2.9461	2.3769	3.9438	4.3213	2.3243	2.5253
recall@50	**7.6148**	3.5811	3.8267	6.0284	5.0657	2.8173	4.1777	4.5599	3.2173	4.9634	6.4679	3.5878	3.5136
recall@100	9.2270	8.5839	8.6231	8.4264	9.1708	7.1184	9.0129	9.4397	7.3718	8.9361	**9.5915**	8.0896	7.6672
ndcg@5	**1.4764**	0.9593	1.0555	1.1867	1.0528	0.6405	0.9988	1.3356	0.9991	1.0935	1.0553	0.9359	0.9105
ndcg@10	**1.9867**	1.1310	1.2805	1.5257	1.4087	0.8242	1.2674	1.6110	1.1861	1.3354	1.2824	1.0682	1.0879
ndcg@30	**3.0016**	1.4666	1.5623	2.4040	2.1343	1.2250	1.8631	2.0355	1.5637	2.1783	2.1876	1.4788	1.4903
ndcg@50	**3.7044**	1.8322	1.9700	2.8513	2.5330	1.4781	2.2273	2.6932	1.8355	2.5507	2.7545	1.8402	1.8013
ndcg@100	**4.1850**	3.0413	3.2487	3.4717	3.6183	2.5676	3.5406	3.9128	2.8803	3.5253	3.5540	2.9559	2.8733
Process_data_time													
seconds	286.0810	276.4867	**264.1901**	264.8814	266.7158	267.9199	269.1589	279.2522	280.8876	281.2558	282.3455	283.0533	284.3426
hrs	0:04:46	0:04:36	**0:04:24**	0:04:25	0:04:27	0:04:28	0:04:29	0:04:39	0:04:41	0:04:41	0:04:42	0:04:43	0:04:44

Y_train_dim	(4000, 6651)	Label Cardinality	4.4873	Training Samples/Label	0.6014
Y_test_dim	(1000, 6651)	Label Density	6.7467E-04		

ตาราง 4.2 : เปรียบเทียบผลข้อมูล Linear 1-5000 ชุดทดสอบ สำหรับสัดส่วนการลดมิติ 0.8 ระหว่างแบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับ แบบจำลองตั้งต้น

Random Method	Linear 1-5000	Testing Result											
XMLC-PAO Ratio	org	XMLC-PAO-1 : 0.8						XMLC-PAO-2 : 0.8					
Adam Learning	0.003	0.0015	0.003	0.006	0.008	0.01	0.02	0.0015	0.003	0.006	0.008	0.01	0.02
mrr	0.0365	0.0211	0.0275	0.0206	0.0204	0.0260	0.0296	0.0249	0.0236	0.0239	0.0244	0.0329	**0.0369**
recall@5	**2.0891**	0.7302	1.2239	0.4723	0.9411	1.0991	1.2012	1.3327	0.9153	1.0961	1.0160	1.5445	2.0224
recall@10	3.0334	1.0579	2.0901	0.9552	1.1972	1.4757	1.4960	2.0059	1.2653	1.5347	1.4242	2.1845	**3.0396**
recall@30	5.4200	1.9511	3.5693	1.5763	1.5346	2.2081	2.1015	3.4993	2.1321	2.5930	2.6951	3.7227	**5.9568**
recall@50	7.6148	3.1265	4.7107	2.0826	2.5945	3.6163	4.1894	5.7564	2.8948	3.6950	3.7178	4.9089	**7.9776**
recall@100	9.2270	7.1954	8.3646	6.9748	8.0288	7.6526	8.4137	9.2408	8.1906	8.2971	6.3807	8.7190	**12.1814**
ndcg@5	1.4764	0.6763	0.9801	0.5398	0.7944	0.9539	1.0798	0.9739	0.7478	0.8664	0.8732	1.2506	**1.5471**
ndcg@10	1.9867	0.8285	1.3737	0.7574	0.9311	1.1659	1.2489	1.2989	0.9343	1.0972	1.1071	1.6014	**2.0365**
ndcg@30	3.0016	1.1837	1.9887	1.0540	1.1287	1.5422	1.5940	1.9370	1.3009	1.5778	1.6447	2.2599	**3.1599**
ndcg@50	3.7044	1.5151	2.3855	1.2272	1.4561	1.9583	2.2241	2.5836	1.5293	1.9391	1.9793	2.6945	**3.7483**
ndcg@100	4.1850	2.5199	3.3373	2.4092	2.7453	3.0439	3.3873	3.4483	2.8137	3.1496	2.7227	3.6962	**4.7933**
Process_data_time													
seconds	286.0810	**271.3268**	272.5074	274.0789	274.4366	276.4196	276.3945	288.9709	290.2965	291.2522	292.2474	293.8584	294.5012
hrs	0:04:46	**0:04:31**	0:04:33	0:04:34	0:04:34	0:04:36	0:04:36	0:04:49	0:04:50	0:04:51	0:04:52	0:04:54	0:04:55

Y_train_dim	(4000, 6651)	Label Cardinality	4.4873	Training Samples/Label	0.6014
Y_test_dim	(1000, 6651)	Label Density	6.7467E-04		

ตาราง 4.3 : เปรียบเทียบผลข้อมูล Linear 1-5000 ชุดทดสอบ สำหรับสัดส่วนการลดมิติ 1.0 ระหว่างแบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับ แบบจำลองตั้งต้น

Random Method	Linear 1-5000	Testing Result											
XMLC-PAO Ratio	org	XMLC-PAO-1 : 1.0						XMLC-PAO-2 : 1.0					
Adam Learning	0.003	0.0015	0.003	0.006	0.008	0.01	0.02	0.0015	0.003	0.006	0.008	0.01	0.02
mrr	0.0365	0.0287	0.0311	0.0345	0.0312	0.0253	0.0339	0.0294	0.0234	0.0284	0.0281	0.0274	0.0326
recall@5	2.0891	1.5724	1.7655	1.2418	1.1405	1.4874	1.9366	1.3385	0.5074	1.5986	1.4111	1.1404	1.8830
recall@10	3.0334	2.0420	2.2721	2.4064	2.2361	2.2144	2.7226	1.9416	1.0538	2.2711	1.8592	1.6388	2.3552
recall@30	5.4200	3.9073	3.6397	4.3699	4.3218	3.2946	4.5393	3.3386	2.1229	3.9887	3.4070	2.4071	4.4805
recall@50	7.6148	5.3588	5.3293	7.0437	5.5282	4.5212	5.8158	4.8029	3.9074	4.9535	5.0221	4.1000	5.8318
recall@100	9.2270	9.6981	9.3404	11.1003	9.0054	7.8933	9.5335	8.6015	9.3783	9.5006	8.9596	8.2992	8.6003
ndcg@5	1.4764	1.1466	1.3421	1.2041	1.1523	1.0408	1.4324	1.0651	0.6227	1.1361	1.0689	1.0022	1.4064
ndcg@10	1.9867	1.3677	1.6222	1.7260	1.7168	1.3694	1.8109	1.3663	0.9053	1.4858	1.3235	1.2644	1.6508
ndcg@30	3.0016	2.1431	2.2527	2.5880	2.5576	1.8228	2.5939	1.9352	1.3810	2.2741	1.9552	1.6790	2.5785
ndcg@50	3.7044	2.6223	2.7497	3.2864	2.9650	2.2285	3.0570	2.3718	2.0406	2.6217	2.4093	2.2023	3.0129
ndcg@100	4.1850	3.7202	3.8140	4.2777	3.8958	3.1506	4.0541	3.4289	3.3266	3.7890	3.4488	3.3575	3.7479
Process_data_time													
seconds	286.0810	278.7729	281.3054	281.3162	282.5347	284.7627	284.4292	299.4775	300.5688	301.4037	302.7960	305.0587	304.0337
hrs	0:04:46	0:04:39	0:04:41	0:04:41	0:04:43	0:04:45	0:04:44	0:04:59	0:05:01	0:05:01	0:05:03	0:05:05	0:05:04

Y_train_dim	(4000, 6651)	Label Cardinality	4.4873	Training Samples/Label	0.6014
Y_test_dim	(1000, 6651)	Label Density	6.7467E-04		

ตาราง 4.4 : เปรียบเทียบผลข้อมูล Linear 201-5200 ชุดทดสอบ สำหรับสัดส่วนการลดมิติ 0.5 ระหว่างแบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับ แบบจำลองตั้งต้น

Random Method	Linear 201-5200	Testing Result											
XMLC-PAO Ratio	org	XMLC-PAO-1 : 0.5						XMLC-PAO-2 : 0.5					
Adam Learning	0.003	0.0015	0.003	0.006	0.008	0.01	0.02	0.0015	0.003	0.006	0.008	0.01	0.02
mrr	0.0472	0.0366	0.0387	0.0292	0.0437	0.0188	0.0427	0.0421	**0.0473**	0.0199	0.0216	0.0461	0.0364
recall@5	1.5414	1.5996	1.7202	1.3143	**2.4568**	0.6727	1.3593	1.9325	1.9139	0.5995	0.6274	2.3332	1.1992
recall@10	**3.3215**	1.9357	2.3895	1.5412	2.8282	0.8718	1.9208	2.3472	3.0482	0.6641	0.6635	3.1440	1.7958
recall@30	5.3865	2.6067	4.0232	2.5928	3.8394	1.2322	3.8025	3.6165	5.0904	1.1771	1.0669	**5.5442**	3.0533
recall@50	7.4646	3.3420	5.6376	3.7224	5.3906	1.4778	6.3574	4.6485	6.3568	1.4925	1.3995	**8.3470**	3.8779
recall@100	10.5961	8.4244	9.6906	8.6731	8.7754	5.6959	10.7711	9.0241	9.5530	7.1388	3.3608	**11.4456**	9.4588
ndcg@5	1.7031	1.4099	1.5333	1.1520	1.9330	0.6208	1.5287	1.6735	1.8100	0.6437	0.5903	**2.0120**	1.3174
ndcg@10	**2.3793**	1.5942	1.8119	1.2883	2.1534	0.7261	1.8371	1.9068	2.3445	0.6781	0.6305	2.3767	1.6575
ndcg@30	3.1913	1.8859	2.4104	1.7267	2.6572	0.8824	2.5969	2.4671	**3.2244**	0.8727	0.8190	3.2013	2.2401
ndcg@50	3.9156	2.1600	2.9406	2.1736	3.1911	1.0063	3.3640	2.8142	3.6777	1.0142	0.9723	**3.9317**	2.5425
ndcg@100	**4.8292**	3.3171	3.9249	3.3467	4.1037	2.0866	4.4495	3.8698	4.4807	2.3501	1.6550	4.7106	3.8328
Process_data_time													
seconds	298.4902	274.6650	275.9238	277.4712	277.4948	279.5860	280.7333	287.7428	276.8370	277.8015	277.1003	**274.6248**	277.6804
hrs	0:04:58	0:04:35	0:04:36	0:04:37	0:04:37	0:04:40	0:04:41	0:04:48	0:04:37	0:04:38	0:04:37	**0:04:35**	0:04:38

Y_train_dim (4000, 6646) **Label Cardinality** 4.3305 **Training Samples/Label** 0.6019
Y_test_dim (1000, 6646) **Label Density** 6.5159E-04

ตาราง 4.5 : เปรียบเทียบผลข้อมูล Linear 201-5200 ชุดทดสอบ สำหรับสัดส่วนการลดมิติ 0.8 ระหว่างแบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับ แบบจำลองตั้งต้น

Random Method	Linear 201-5200	Testing Result											
XMLC-PAO Ratio	org	XMLC-PAO-1 : 0.8						XMLC-PAO-2 : 0.8					
Adam Learning	0.003	0.0015	0.003	0.006	0.008	0.01	0.02	0.0015	0.003	0.006	0.008	0.01	0.02
mrr	0.0472	0.0424	0.0454	0.0457	0.0410	0.0298	0.0328	0.0406	0.0307	**0.0474**	0.0435	0.0202	0.0418
recall@5	1.5414	2.1027	2.2041	2.3342	2.3098	0.6599	1.5639	1.4363	1.1772	**2.5130**	2.3043	0.4144	1.8894
recall@10	3.3215	2.9704	2.9728	3.0029	2.6957	1.6884	1.8581	2.2961	1.3339	**3.3650**	3.0570	1.0115	2.5567
recall@30	5.3865	4.1876	5.0489	5.3275	4.1748	2.3919	3.0339	4.9566	1.6479	**6.1234**	4.5028	1.3534	3.9095
recall@50	7.4646	5.5512	7.0595	8.2285	5.9408	2.7441	3.6712	7.0811	2.2921	**8.6979**	5.6345	1.7283	6.3257
recall@100	10.5961	8.8433	10.9263	**13.1604**	8.8745	7.2335	7.2775	11.3495	6.3969	11.4867	9.4163	6.1787	10.6778
ndcg@5	1.7031	1.8236	1.8172	2.0002	1.7983	0.7978	1.3513	1.4708	1.1093	**2.0108**	1.9433	0.4463	1.7182
ndcg@10	2.3793	2.2001	2.1702	2.3742	2.0257	1.2026	1.5060	1.9339	1.1920	**2.4153**	2.2824	0.6927	2.0280
ndcg@30	3.1913	2.7086	2.9274	3.3285	2.5716	1.5848	1.9581	2.9240	1.3572	**3.4909**	2.8031	0.8907	2.5738
ndcg@50	3.9156	3.1886	3.6098	4.1128	3.1059	1.7548	2.2224	3.4631	1.5775	**4.2641**	3.1837	1.0375	3.2924
ndcg@100	4.8292	4.1035	4.6045	**5.2551**	3.9531	2.8393	3.1868	4.4746	2.6168	5.0149	4.2724	2.0983	4.4394
Process_data_time													
seconds	298.4902	283.5981	284.9692	285.9180	286.8207	288.2754	289.2524	282.9559	283.9396	**281.5435**	283.6406	283.0063	285.3268
hrs	0:04:58	0:04:44	0:04:45	0:04:46	0:04:47	0:04:48	0:04:49	0:04:43	0:04:44	**0:04:42**	0:04:44	0:04:43	0:04:45

Y_train_dim (4000, 6646) **Label Cardinality** 4.3305 **Training Samples/Label** 0.6019
Y_test_dim (1000, 6646) **Label Density** 6.5159E-04

ตาราง 4.7 : เปรียบเทียบผลข้อมูล Linear 201-5200 ชุดทดสอบ สำหรับสัดส่วนการลดมิติ 1.0 ระหว่างแบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับ แบบจำลองตั้งต้น

Random Method	Linear 801-5800	Testing Result											
XMLC-PAO Ratio	org	XMLC-PAO-1 : 0.5						XMLC-PAO-2 : 0.5					
Adam Learning	0.003	0.0015	0.003	0.006	0.008	0.01	0.02	0.0015	0.003	0.006	0.008	0.01	0.02
mrr	**0.0469**	0.0423	0.0346	0.0443	0.0373	0.0334	0.0413	0.0416	0.0354	0.0430	0.0444	0.0394	0.0389
recall@5	2.4902	1.8682	1.4484	**2.5191**	1.9495	1.1598	2.2918	1.6246	1.5569	2.0287	2.3430	1.6764	1.9685
recall@10	3.2302	2.9538	1.7857	**3.3512**	2.5913	1.9144	2.6646	2.4352	2.2895	2.9991	2.8405	2.7367	2.8631
recall@30	**5.9076**	4.4974	2.8571	5.3572	3.4286	2.6913	3.3229	4.4353	3.3675	3.9625	4.1072	5.1626	5.7999
recall@50	7.7635	5.1414	3.5863	7.2069	4.9669	4.6526	3.8788	6.4610	4.6917	5.0361	5.1001	7.2115	**8.2743**
recall@100	10.3164	9.0479	7.8305	9.8688	8.4027	9.0130	6.5868	9.1858	8.1954	8.5874	10.1420	12.0237	**12.1090**
ndcg@5	2.0496	1.7313	1.3068	**2.1079**	1.6689	1.1711	1.9453	1.5910	1.3848	1.7908	1.9713	1.5893	1.5851
ndcg@10	2.4330	2.2245	1.4804	**2.4738**	1.9652	1.5034	2.1672	1.9934	1.7280	2.2311	2.2432	2.0128	2.0120
ndcg@30	**3.3850**	2.8075	1.8782	3.1743	2.3569	1.8743	2.4867	2.7475	2.1256	2.6594	2.7479	2.8351	3.0702
ndcg@50	**3.9415**	3.0683	2.1074	3.6425	2.7867	2.5035	2.6891	3.2879	2.5028	3.0028	3.0536	3.4015	3.7529
ndcg@100	**4.7976**	4.0344	3.1489	4.3149	3.6297	3.6136	3.4032	3.9935	3.3648	3.8875	4.2450	4.5374	4.7099
Process_data_time													
seconds	283.4980	278.3665	267.7825	267.5885	**265.9119**	268.1987	269.4404	298.2491	289.3050	290.1026	291.0396	292.9776	293.5673
hrs	0:04:43	0:04:38	0:04:28	0:04:28	**0:04:26**	0:04:28	0:04:29	0:04:58	0:04:49	0:04:50	0:04:51	0:04:53	0:04:54

Y_train_dim	(4000, 6682)	**Label Cardinality**	3.9830	**Training Samples/Label**	0.5986
Y_test_dim	(1000, 6682)	**Label Density**	5.9608E-04		

ตาราง 4.7 : เปรียบเทียบผลข้อมูล Linear 801-5800 ชุดทดสอบ สำหรับสัดส่วนการลดมิติ 0.5 ระหว่างแบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับ แบบจำลองตั้งต้น

Random Method	Linear 801-5800	Testing Result											
XMLC-PAO Ratio	org	XMLC-PAO-1 : 0.5						XMLC-PAO-2 : 0.5					
Adam Learning	0.003	0.0015	0.003	0.006	0.008	0.01	0.02	0.0015	0.003	0.006	0.008	0.01	0.02
mrr	**0.0469**	0.0423	0.0346	0.0443	0.0373	0.0334	0.0413	0.0416	0.0354	0.0430	0.0444	0.0394	0.0389
recall@5	2.4902	1.8682	1.4484	**2.5191**	1.9495	1.1598	2.2918	1.6246	1.5569	2.0287	2.3430	1.6764	1.9685
recall@10	3.2302	2.9538	1.7857	**3.3512**	2.5913	1.9144	2.6646	2.4352	2.2895	2.9991	2.8405	2.7367	2.8631
recall@30	**5.9076**	4.4974	2.8571	5.3572	3.4286	2.6913	3.3229	4.4353	3.3675	3.9625	4.1072	5.1626	5.7999
recall@50	7.7635	5.1414	3.5863	7.2069	4.9669	4.6526	3.8788	6.4610	4.6917	5.0361	5.1001	7.2115	**8.2743**
recall@100	10.3164	9.0479	7.8305	9.8688	8.4027	9.0130	6.5868	9.1858	8.1954	8.5874	10.1420	12.0237	**12.1090**
ndcg@5	2.0496	1.7313	1.3068	**2.1079**	1.6689	1.1711	1.9453	1.5910	1.3848	1.7908	1.9713	1.5893	1.5851
ndcg@10	2.4330	2.2245	1.4804	**2.4738**	1.9652	1.5034	2.1672	1.9934	1.7280	2.2311	2.2432	2.0128	2.0120
ndcg@30	**3.3850**	2.8075	1.8782	3.1743	2.3569	1.8743	2.4867	2.7475	2.1256	2.6594	2.7479	2.8351	3.0702
ndcg@50	**3.9415**	3.0683	2.1074	3.6425	2.7867	2.5035	2.6891	3.2879	2.5028	3.0028	3.0536	3.4015	3.7529
ndcg@100	**4.7976**	4.0344	3.1489	4.3149	3.6297	3.6136	3.4032	3.9935	3.3648	3.8875	4.2450	4.5374	4.7099
Process_data_time													
seconds	283.4980	278.3665	267.7825	267.5885	**265.9119**	268.1987	269.4404	298.2491	289.3050	290.1026	291.0396	292.9776	293.5673
hrs	0:04:43	0:04:38	0:04:28	0:04:28	**0:04:26**	0:04:28	0:04:29	0:04:58	0:04:49	0:04:50	0:04:51	0:04:53	0:04:54

Y_train_dim (4000, 6682) **Label Cardinality** 3.9830 **Training Samples/Label** 0.5986
Y_test_dim (1000, 6682) **Label Density** 5.9608E-04

ตาราง 4.8 : เปรียบเทียบผลข้อมูล Linear 801-5800 ชุดทดสอบ สำหรับสัดส่วนการลดมิติ 0.8 ระหว่างแบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับ แบบจำลองตั้งต้น

Random Method	Linear 801-5800	Testing Result											
XMLC-PAO Ratio	org	XMLC-PAO-1 : 0.8						XMLC-PAO-2 : 0.8					
Adam Learning	0.003	0.0015	0.003	0.006	0.008	0.01	0.02	0.0015	0.003	0.006	0.008	0.01	0.02
mrr	**0.0469**	0.0405	0.0341	0.0443	0.0330	0.0414	0.0377	0.0311	0.0389	0.0420	0.0451	0.0428	0.0346
recall@5	2.4902	1.9635	1.5774	1.6868	1.2340	1.9230	1.8544	1.4530	1.4190	1.8198	**2.5657**	2.4068	1.2252
recall@10	**3.2302**	2.5099	1.9769	2.9109	1.7370	2.8946	2.5764	1.6677	2.2507	2.6897	3.2103	3.2144	1.7427
recall@30	**5.9076**	4.2151	2.5235	4.9309	3.1263	4.2207	4.3789	2.4169	3.5186	5.7722	5.4566	5.4483	3.1589
recall@50	7.7635	6.0409	3.6578	6.2509	5.8714	5.5846	6.0397	3.2159	4.9845	7.3019	**7.9920**	7.8459	3.8710
recall@100	10.3164	8.5510	8.2139	10.9432	9.1321	9.5044	9.3541	6.1989	7.9497	11.0270	**12.5838**	12.2290	8.5201
ndcg@5	2.0496	1.7587	1.4324	1.7053	1.2452	1.6954	1.6378	1.1495	1.3890	1.7211	**2.1232**	1.9672	1.1264
ndcg@10	2.4330	2.0726	1.6311	2.2359	1.5210	2.1350	1.9611	1.2646	1.7516	2.1521	**2.4530**	2.3334	1.3942
ndcg@30	**3.3850**	2.8063	1.8630	2.9586	2.0983	2.6427	2.6091	1.5625	2.2848	3.0938	3.1737	3.0900	1.9356
ndcg@50	**3.9415**	3.4089	2.1621	3.3576	2.8405	3.0076	3.0505	1.7971	2.7544	3.5104	3.8500	3.7450	2.1835
ndcg@100	4.7976	4.0474	3.2656	4.5412	3.6371	3.9747	3.8758	2.6629	3.5448	4.4084	**4.9226**	4.8632	3.3610
Process_data_time													
seconds	283.4980	**271.7933**	271.8287	273.0143	272.5572	276.1511	276.1874	298.3517	299.2540	300.3995	301.2700	302.5294	303.4267
hrs	0:04:43	**0:04:32**	0:04:32	0:04:33	0:04:33	0:04:36	0:04:36	0:04:58	0:04:59	0:05:00	0:05:01	0:05:03	0:05:03

Y_train_dim (4000, 6682) **Label Cardinality** 3.9830 **Training Samples/Label** 0.5986
Y_test_dim (1000, 6682) **Label Density** 5.9608E-04

ตาราง 4.9 : เปรียบเทียบผลข้อมูล Linear 801-5800 ชุดทดสอบ สำหรับสัดส่วนการลดมิติ 1.0 ระหว่างแบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับ แบบจำลองตั้งต้น

Random Method	Linear 801-5800	Testing Result											
XMLC-PAO Ratio	org	XMLC-PAO-1 : 1.0						XMLC-PAO-2 : 1.0					
Adam Learning	0.003	0.0015	0.003	0.006	0.008	0.01	0.02	0.0015	0.003	0.006	0.008	0.01	0.02
mrr	0.0469	0.0446	0.0307	0.0361	0.0408	0.0421	0.0393	0.0363	0.0431	0.0440	0.0342	0.0382	0.0446
recall@5	2.4902	2.1434	1.3286	1.7459	1.9269	1.8075	1.9314	1.3533	1.8091	2.3137	1.0746	1.1749	1.9970
recall@10	3.2302	2.6965	1.5657	2.0758	2.6841	2.9841	2.4452	1.8643	2.9035	3.1067	1.5173	2.2096	2.6903
recall@30	5.9076	4.0302	2.4455	2.7375	4.0731	4.5166	5.0301	3.2859	4.2724	4.5969	2.7161	3.4051	5.8812
recall@50	7.7635	5.1367	2.9942	3.2901	5.5213	6.3290	6.9570	3.9201	5.6322	6.1747	5.5960	4.4859	8.0711
recall@100	10.3164	9.1956	5.4523	6.7586	9.7110	9.3954	10.9417	8.1359	10.5185	10.6167	9.3135	8.0111	12.1950
ndcg@5	2.0496	1.8511	1.1394	1.5194	1.7151	1.7038	1.6279	1.3696	1.7462	1.9661	1.1390	1.2233	1.8396
ndcg@10	2.4330	2.1365	1.2658	1.7141	2.0861	2.2210	1.8944	1.6219	2.1768	2.3665	1.4303	1.6416	2.1975
ndcg@30	3.3850	2.6352	1.5926	1.9914	2.6213	2.8126	2.7144	2.0928	2.7068	2.9237	1.9925	2.0573	3.2994
ndcg@50	3.9415	2.9501	1.7700	2.1575	3.0344	3.2939	3.2326	2.2926	3.0929	3.3719	2.8002	2.3369	3.8173
ndcg@100	4.7976	3.9196	2.5024	3.0430	3.9789	4.0424	4.1905	3.2962	4.2578	4.4328	3.7367	3.2287	4.7815
Process_data_time													
seconds	283.4980	278.7431	281.0906	279.1697	279.4847	281.9901	280.9967	307.7984	310.2095	309.8961	311.1111	311.4987	313.5110
hrs	0:04:43	0:04:39	0:04:41	0:04:39	0:04:39	0:04:42	0:04:41	0:05:08	0:05:10	0:05:10	0:05:11	0:05:11	0:05:14

Y_train_dim	(4000, 6682)	Label Cardinality	3.9830	Training Samples/Label	0.5986
Y_test_dim	(1000, 6682)	Label Density	5.9608E-04		

ตาราง 4.10 : เปรียบเทียบผลข้อมูล Linear 5001-10000 ชุดทดสอบ สำหรับสัดส่วนการลดมิติ 0.5 ระหว่างแบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับ แบบจำลองตั้งต้น

Random Method	Linear 5001-10000	Testing Result											
XMLC-PAO Ratio	org	XMLC-PAO-1 : 0.5						XMLC-PAO-2 : 0.5					
Adam Learning	0.003	0.0015	0.003	0.006	0.008	0.01	0.02	0.0015	0.003	0.006	0.008	0.01	0.02
mrr	**0.0397**	0.0363	0.0313	0.0362	0.0392	0.0360	0.0371	0.0387	0.0324	0.0351	0.0359	0.0349	0.0370
recall@5	1.5283	1.5532	1.3891	**1.7496**	1.4561	1.4479	1.6183	1.7015	1.4669	1.4105	1.5943	1.5303	1.6421
recall@10	**2.7876**	2.2560	1.8946	2.5672	2.3825	2.2041	2.2901	2.3964	2.3035	2.5182	2.3231	2.3719	2.2804
recall@30	5.7617	4.3181	3.6349	**6.7613**	4.9553	4.4081	4.7210	5.3598	4.7794	4.9165	4.8526	4.6227	4.3116
recall@50	7.7477	5.9145	5.6100	**9.2918**	6.9149	6.4325	7.4145	7.2539	5.9932	7.0804	6.7690	5.6362	6.4574
recall@100	10.7601	9.4513	8.7663	**13.3864**	9.9367	10.0872	9.6532	11.0120	8.6158	10.2883	9.2426	8.5645	9.4503
ndcg@5	1.5317	1.5144	1.3246	**1.6322**	1.5189	1.4183	1.5152	1.6045	1.3085	1.3394	1.4841	1.4597	1.5721
ndcg@10	**2.0432**	1.7895	1.5211	2.0021	1.8989	1.7300	1.8131	1.9044	1.6252	1.7667	1.8109	1.7919	1.8349
ndcg@30	2.9328	2.4083	2.0556	**3.2470**	2.7342	2.3979	2.5466	2.7411	2.4035	2.4968	2.5214	2.4938	2.4783
ndcg@50	3.4034	2.8198	2.5418	**3.8833**	3.2125	2.8775	3.1756	3.2100	2.7124	3.0509	2.9782	2.7507	2.9929
ndcg@100	4.0708	3.5761	3.2354	**4.7760**	3.8791	3.6902	3.6732	4.0063	3.3106	3.7260	3.5355	3.3879	3.6439
Process_data_time													
seconds	276.5084	272.7186	263.1741	263.3456	263.2724	**261.5893**	262.5668	282.1414	273.1717	272.5722	273.8270	275.2396	276.1482
hrs	0:04:37	0:04:33	0:04:23	0:04:23	0:04:23	**0:04:22**	0:04:23	0:04:42	0:04:33	0:04:33	0:04:34	0:04:35	0:04:36

Y_train_dim (4000, 6937) **Label Cardinality** 3.1175 **Training Samples/Label** 0.5766
Y_test_dim (1000, 6937) **Label Density** 4.4940E-04

ตาราง 4.11 : เปรียบเทียบผลข้อมูล Linear 5001-10000 ชุดทดสอบ สำหรับสัดส่วนการลดมิติ 0.8 ระหว่างแบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับ แบบจำลองตั้งต้น

Random Method	Linear 5001-10000	Testing Result											
XMLC-PAO Ratio	org	XMLC-PAO-1 : 0.8						XMLC-PAO-2 : 0.8					
Adam Learning	0.003	0.0015	0.003	0.006	0.008	0.01	0.02	0.0015	0.003	0.006	0.008	0.01	0.02
mrr	**0.0397**	0.0355	0.0380	0.0375	0.0359	0.0378	0.0362	0.0335	0.0381	0.0384	0.0313	0.0361	0.0377
recall@5	1.5283	1.4825	1.6650	1.7681	1.5681	1.7223	1.7571	1.7143	**1.8412**	1.6237	1.5222	1.5593	1.6131
recall@10	2.7876	2.5519	2.7197	**2.8778**	2.5293	2.6348	2.7023	2.8466	2.8029	2.5432	2.3699	2.5075	2.6043
recall@30	**5.7617**	5.0031	5.1084	5.1199	5.2439	5.1998	4.4655	5.6216	5.1806	4.5586	4.9459	4.6273	5.2921
recall@50	**7.7477**	6.5244	7.5383	6.7005	7.0527	6.4562	7.1158	7.5691	6.9329	5.8238	6.1937	6.6653	6.5145
recall@100	10.7601	9.5842	11.2082	9.7638	10.4683	9.5964	10.3881	11.2493	**11.4955**	9.0110	8.6181	9.4111	9.9342
ndcg@5	1.5317	1.4206	1.5365	1.6071	1.4540	1.6087	1.6165	1.4527	**1.6247**	1.6140	1.2822	1.4642	1.5492
ndcg@10	2.0432	1.8856	1.9887	2.0443	1.8779	1.9710	1.9968	1.9196	**2.0533**	1.9606	1.6704	1.8753	1.9639
ndcg@30	**2.9328**	2.6292	2.7330	2.7205	2.6695	2.7372	2.5460	2.7164	2.7869	2.5625	2.4472	2.4899	2.7505
ndcg@50	**3.4034**	2.9825	3.3094	3.1161	3.0800	3.0603	3.1481	3.1876	3.2207	2.8555	2.7580	2.9738	3.0629
ndcg@100	4.0708	3.5990	4.0779	3.8020	3.8278	3.7327	3.8302	3.9501	**4.1730**	3.5781	3.2724	3.5558	3.7956
Process_data_time													
seconds	276.5084	**265.0896**	265.5832	267.0533	266.6951	269.1758	269.6548	282.4170	284.3386	285.5109	284.3518	285.7659	287.2942
hrs	0:04:37	**0:04:25**	0:04:26	0:04:27	0:04:27	0:04:29	0:04:30	0:04:42	0:04:44	0:04:46	0:04:44	0:04:46	0:04:47

Y_train_dim (4000, 6937) **Label Cardinality** 3.1175 **Training Samples/Label** 0.5766
Y_test_dim (1000, 6937) **Label Density** 4.4940E-04

ตาราง 4.12 : เปรียบเทียบผลข้อมูล Linear 5001-10000 ชุดทดสอบ สำหรับสัดส่วนการลดมิติ 1.0 ระหว่างแบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับ แบบจำลองตั้งต้น

Random Method	Linear 5001-10000	Testing Result											
XMLC-PAO Ratio	org	XMLC-PAO-1 : 1.0						XMLC-PAO-2 : 1.0					
Adam Learning	0.003	0.0015	0.003	0.006	0.008	0.01	0.02	0.0015	0.003	0.006	0.008	0.01	0.02
mrr	0.0397	0.0350	**0.0405**	0.0346	0.0361	0.0389	0.0378	0.0336	0.0331	0.0362	0.0369	0.0401	0.0371
recall@5	1.5283	1.5125	1.6089	1.4444	1.7063	**1.8754**	1.4636	1.6841	1.4350	1.6372	1.3650	1.6849	1.4783
recall@10	2.7876	2.4477	2.6398	2.1075	**2.8235**	2.7223	2.4142	2.3942	2.2277	2.4573	2.1067	2.6476	2.4135
recall@30	5.7617	5.5173	5.5276	3.8096	5.7185	4.8889	4.5752	4.3670	4.8218	**5.8817**	4.6111	5.2401	4.6327
recall@50	7.7477	**7.9921**	7.8191	5.4001	7.2414	6.9842	6.9159	6.1095	6.9505	7.1666	5.9510	6.3911	6.6268
recall@100	10.7601	11.2546	**11.6105**	8.8943	10.2783	9.9856	10.0836	8.5490	10.2375	10.2554	9.1276	9.1837	9.9416
ndcg@5	1.5317	1.4238	1.6229	1.4032	1.4533	**1.6896**	1.4803	1.4734	1.2954	1.4995	1.4262	1.6599	1.4666
ndcg@10	2.0432	1.7920	2.0221	1.6857	1.9326	2.0362	1.8508	1.7731	1.6371	1.8616	1.7161	**2.0546**	1.8437
ndcg@30	2.9328	2.6067	**2.9372**	2.1864	2.7874	2.7115	2.5365	2.3801	2.3829	2.7872	2.4954	2.8455	2.5493
ndcg@50	3.4034	3.1839	**3.5013**	2.5712	3.1650	3.2213	3.0929	2.7980	2.8752	3.1138	2.8337	3.1420	3.0620
ndcg@100	4.0708	3.8866	**4.3287**	3.2996	3.8516	3.8347	3.7214	3.3361	3.5886	3.7912	3.5325	3.7885	3.7464
Process_data_time													
seconds	276.5084	**271.6128**	273.7317	274.2819	274.4355	276.5168	275.5372	288.4194	291.7164	293.5335	293.1274	294.6327	295.4843
hrs	0:04:37	**0:04:32**	0:04:34	0:04:34	0:04:34	0:04:37	0:04:36	0:04:48	0:04:52	0:04:54	0:04:53	0:04:55	0:04:55

Y_train_dim	(4000, 6937)	**Label Cardinality**	3.1175	**Training Samples/Label**	0.5766
Y_test_dim	(1000, 6937)	**Label Density**	4.4940E-04		

ตาราง 4.13 : เปรียบเทียบผลข้อมูล Linear 10001-15000 ชุดทดสอบ สำหรับสัดส่วนการลดมิติ 0.5 ระหว่างแบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับ แบบจำลองตั้งต้น

Random Method	Linear 10001-15000	Testing Result											
XMLC-PAO Ratio	org	XMLC-PAO-1 : 0.5						XMLC-PAO-2 : 0.5					
Adam Learning	0.003	0.0015	0.003	0.006	0.008	0.01	0.02	0.0015	0.003	0.006	0.008	0.01	0.02
mrr	0.0218	0.0168	0.0153	0.0193	0.0190	0.0189	0.0197	0.0205	0.0186	0.0181	0.0185	0.0212	0.0195
recall@5	0.9258	0.8642	0.6225	1.3183	0.9058	0.9633	0.9742	0.9325	0.9700	1.4192	1.0142	1.5000	0.9517
recall@10	2.1175	1.4267	1.5517	1.8833	1.3842	1.7394	1.9067	1.7900	1.5242	2.0717	1.9900	2.3575	1.8017
recall@30	4.9253	3.5758	3.9083	4.6083	3.5650	3.7128	4.0353	4.3375	3.5142	4.1261	3.7400	4.4483	4.1418
recall@50	7.5162	5.2767	6.2729	5.7317	5.6694	5.4978	5.2878	7.1858	5.5886	5.9961	5.5133	5.9671	5.3468
recall@100	12.0149	7.6694	9.2497	8.7759	8.3646	7.9794	7.9030	9.9504	8.2460	8.9361	9.1672	9.4880	7.9054
ndcg@5	0.7735	0.7207	0.4941	0.8980	0.7476	0.8353	0.8627	0.7489	0.7717	1.0606	0.7643	1.0716	0.7583
ndcg@10	1.2230	0.9492	0.8726	1.1590	0.9493	1.1613	1.2280	1.0907	1.0140	1.3319	1.1946	1.4073	1.1005
ndcg@30	2.0927	1.5629	1.5698	1.9903	1.5615	1.7512	1.8509	1.8580	1.6284	1.9279	1.6914	2.0295	1.8203
ndcg@50	2.6741	1.9666	2.1004	2.2802	2.0643	2.1663	2.1515	2.5162	2.1035	2.3379	2.0933	2.3835	2.0999
ndcg@100	3.6189	2.4870	2.7001	2.8996	2.6359	2.6708	2.6604	3.0908	2.6535	2.9242	2.8364	3.1076	2.6215
Process_data_time													
seconds	278.5305	259.2918	259.3344	259.2489	259.2999	261.0926	261.2234	299.6653	286.9303	287.9926	289.6439	291.2047	292.2794
hrs	0:04:39	0:04:19	0:04:19	0:04:19	0:04:19	0:04:21	0:04:21	0:05:00	0:04:47	0:04:48	0:04:50	0:04:51	0:04:52

Y_train_dim	(4000, 7079)	Label Cardinality	2.7965	Training Samples/Label	0.5651
Y_test_dim	(1000, 7079)	Label Density	3.9504E-04		

ตาราง 4.14 : เปรียบเทียบผลข้อมูล Linear 10001-15000 ชุดทดสอบ สำหรับสัดส่วนการลดมิติ 0.8 ระหว่างแบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับ แบบจำลองดั้งต้น

Random Method	Linear 10001-15000	Testing Result											
XMLC-PAO Ratio	org	XMLC-PAO-1 : 0.8						XMLC-PAO-2 : 0.8					
Adam Learning	0.003	0.0015	0.003	0.006	0.008	0.01	0.02	0.0015	0.003	0.006	0.008	0.01	0.02
mrr	0.0218	0.0181	0.0205	0.0198	0.0175	0.0200	0.0174	0.0183	0.0184	0.0203	0.0215	0.0192	0.0188
recall@5	0.9258	0.8025	1.3117	1.0008	0.4475	0.9042	0.8375	1.1117	1.2500	1.0658	1.1208	1.1475	0.9875
recall@10	2.1175	1.6942	2.2278	1.6250	1.1850	1.5392	1.5992	1.9144	2.1493	2.3633	2.0308	1.5250	2.2008
recall@30	4.9253	3.5258	4.3419	4.0242	4.3967	4.5696	4.7122	3.7594	4.1576	4.5550	3.9311	4.3494	4.1595
recall@50	7.5162	4.6792	5.9453	6.2569	6.2425	6.4547	5.9655	5.2496	5.2394	6.0550	5.3637	5.4911	5.2038
recall@100	12.0149	7.2343	7.8965	8.6254	8.5444	8.4710	8.5478	8.2040	7.3654	8.8937	8.8673	9.0813	8.3216
ndcg@5	0.7735	0.6930	0.9225	0.8253	0.4677	0.8701	0.6847	0.8324	0.8906	0.8204	0.9350	0.8717	0.7715
ndcg@10	1.2230	1.0703	1.3014	1.0983	0.7828	1.1050	0.9538	1.1633	1.2571	1.3370	1.3089	1.0453	1.2509
ndcg@30	2.0927	1.6164	1.9360	1.8100	1.7017	2.0095	1.8543	1.6930	1.8480	2.0126	1.9176	1.9106	1.8383
ndcg@50	2.6741	1.8977	2.3206	2.2995	2.1375	2.4490	2.1752	2.0858	2.1059	2.3814	2.2586	2.1932	2.0988
ndcg@100	3.6189	2.4056	2.7544	2.7813	2.6421	2.8531	2.7199	2.6923	2.5616	2.9658	2.9863	2.9105	2.7320
Process_data_time													
seconds	278.5305	263.4875	266.3922	266.5922	268.6146	269.1453	271.0775	297.1155	299.5397	300.4595	301.6624	302.4780	302.5011
hrs	0:04:39	0:04:23	0:04:26	0:04:27	0:04:29	0:04:29	0:04:31	0:04:57	0:05:00	0:05:00	0:05:02	0:05:02	0:05:03

Y_train_dim	(4000, 7079)	Label Cardinality	2.7965	Training Samples/Label	0.5651
Y_test_dim	(1000, 7079)	Label Density	3.9504E-04		

ตาราง 4.15 : เปรียบเทียบผลข้อมูล Linear 10001-15000 ชุดทดสอบ สำหรับสัดส่วนการลดมิติ 1.0 ระหว่างแบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับ แบบจำลองดั้งต้น

Random Method	Linear 10001-15000	Testing Result											
XMLC-PAO Ratio	org	XMLC-PAO-1 : 1.0						XMLC-PAO-2 : 1.0					
Adam Learning	0.003	0.0015	0.003	0.006	0.008	0.01	0.02	0.0015	0.003	0.006	0.008	0.01	0.02
mrr	**0.0218**	0.0187	0.0143	0.0210	0.0201	0.0204	0.0193	0.0157	0.0185	0.0198	0.0165	0.0180	0.0179
recall@5	0.9258	1.2317	0.8375	1.1283	0.9667	**1.4267**	1.0467	0.7517	1.1417	0.6558	0.7100	0.7142	0.8475
recall@10	2.1175	1.9733	1.4325	2.0467	1.6458	1.6892	1.9558	1.2283	1.6675	**2.1475**	1.6083	1.4375	1.5575
recall@30	4.9253	4.5133	3.8600	4.2751	3.8544	4.0769	4.0133	3.8100	3.9794	**5.1743**	3.6313	4.1768	4.4061
recall@50	7.5162	5.9325	5.5778	5.6785	5.5337	5.8580	5.4850	5.6236	5.5387	**7.6043**	5.2749	5.6051	5.6730
recall@100	**12.0149**	8.8443	7.9129	7.7462	8.5255	9.1596	9.1369	8.1241	7.2154	9.5450	7.0750	8.9438	7.8163
ndcg@5	0.7735	0.8837	0.6410	0.9624	0.7936	**1.0306**	0.8532	0.5927	0.9288	0.6444	0.6050	0.6276	0.6849
ndcg@10	1.2230	1.2453	0.8716	**1.3123**	1.0750	1.1513	1.2386	0.8265	1.1469	1.2163	0.9779	0.9170	0.9753
ndcg@30	2.0927	1.9550	1.5633	1.9958	1.7451	1.8795	1.8472	1.6105	1.8819	**2.1009**	1.6031	1.6995	1.7977
ndcg@50	2.6741	2.3008	1.9554	2.3437	2.1518	2.3068	2.1889	2.0253	2.2803	**2.6770**	1.9895	2.0410	2.1067
ndcg@100	**3.6189**	2.9167	2.4550	2.7782	2.7450	2.9708	2.9291	2.5323	2.6358	3.1000	2.3604	2.7687	2.5542
Process_data_time													
seconds	278.5305	272.2162	**271.5172**	273.0762	274.5566	276.2453	278.0829	306.7906	307.0073	308.6738	309.9557	309.4086	312.5458
hrs	0:04:39	0:04:32	**0:04:32**	0:04:33	0:04:35	0:04:36	0:04:38	0:05:07	0:05:07	0:05:09	0:05:10	0:05:09	0:05:13

Y_train_dim (4000, 7079) **Label Cardinality** 2.7965 **Training Samples/Label** 0.5651

Y_test_dim (1000, 7079) **Label Density** 3.9504E-04

ตาราง 4.16 : เปรียบเทียบผลข้อมูล Linear 12201-17200 ชุดทดสอบ สำหรับสัดส่วนการลดมิติ 0.5 ระหว่างแบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับ แบบจำลองตั้งต้น

Random Method	Linear 12201-17200	Testing Result											
XMLC-PAO Ratio	org	XMLC-PAO-1 : 0.5						XMLC-PAO-2 : 0.5					
Adam Learning	0.003	0.0015	0.003	0.006	0.008	0.01	0.02	0.0015	0.003	0.006	0.008	0.01	0.02
mrr	0.0264	0.0218	0.0224	0.0190	0.0214	0.0256	0.0249	0.0229	0.0250	0.0205	0.0236	0.0234	0.0223
recall@5	1.8783	1.3783	1.7983	1.5810	1.6533	1.8676	1.7950	1.5644	1.5718	1.6533	1.5317	1.5383	1.5350
recall@10	2.6817	2.5076	2.3600	1.9526	2.1201	2.3918	2.5073	2.0465	2.2754	2.2400	2.3194	2.7394	2.3079
recall@30	5.0109	4.3279	3.7190	3.9857	3.9773	4.4227	4.8299	4.5648	4.5101	4.1680	4.2101	5.0032	4.2255
recall@50	6.7852	5.4174	4.9761	5.4695	5.3525	5.7475	5.9151	5.9880	5.9981	6.0123	5.3217	6.6619	5.9869
recall@100	10.2794	8.4005	7.6870	8.5550	7.2594	8.0988	7.8542	8.1299	7.8865	8.7255	7.9369	8.6340	8.9907
ndcg@5	1.3120	1.0355	1.3323	1.0645	1.1364	1.4566	1.3326	1.1801	1.2948	1.1424	1.2495	1.1617	1.0894
ndcg@10	1.6578	1.4638	1.5498	1.2289	1.3442	1.6729	1.6318	1.3787	1.5878	1.3675	1.5842	1.6121	1.3934
ndcg@30	2.3575	2.0092	1.9800	1.8534	1.8952	2.3006	2.3049	2.1045	2.2376	1.9193	2.1246	2.2804	1.9644
ndcg@50	2.7991	2.2798	2.2656	2.2018	2.2158	2.6229	2.5503	2.4395	2.5793	2.3291	2.3943	2.6364	2.3609
ndcg@100	3.5170	2.8720	2.8111	2.8127	2.6101	3.0964	2.9425	2.8726	2.9614	2.8974	2.9454	3.0461	2.9700
Process_data_time													
seconds	291.1387	270.0433	270.7243	271.5634	272.5843	273.8946	274.9241	292.5255	279.9617	280.8622	281.3905	282.0465	283.6917
hrs	0:04:51	0:04:30	0:04:31	0:04:32	0:04:33	0:04:34	0:04:35	0:04:53	0:04:40	0:04:41	0:04:41	0:04:42	0:04:44

Y_train_dim	(4000, 7088)	Label Cardinality	2.6660	Training Samples/Label	0.5643
Y_test_dim	(1000, 7088)	Label Density	3.7613E-04		

ตาราง 4.17 : เปรียบเทียบผลข้อมูล Linear 12201-17200 ชุดทดสอบ สำหรับสัดส่วนการลดมิติ 0.8 ระหว่างแบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับ แบบจำลองตั้งต้น

Random Method	Linear 12201-17200	Testing Result											
XMLC-PAO Ratio	org	XMLC-PAO-1 : 0.8						XMLC-PAO-2 : 0.8					
Adam Learning	0.003	0.0015	0.003	0.006	0.008	0.01	0.02	0.0015	0.003	0.006	0.008	0.01	0.02
mrr	0.0264	0.0233	0.0169	0.0227	0.0182	0.0214	0.0231	0.0227	0.0225	0.0231	0.0238	0.0193	0.0219
recall@5	1.8783	1.8333	1.2311	1.6270	1.2367	0.9117	1.6450	1.3917	1.4167	1.8269	1.5750	1.2975	1.5908
recall@10	2.6817	2.3543	2.3911	2.5556	2.1519	2.0683	2.2754	2.3876	2.0454	2.4395	2.0962	2.3785	2.1808
recall@30	5.0109	4.4576	4.7362	4.5696	3.8194	4.3920	4.1562	4.4740	4.0334	4.4684	4.8173	4.5213	4.5637
recall@50	6.7852	5.5904	6.4512	5.7941	5.1637	5.2350	5.8773	5.8450	5.7006	5.6570	6.1052	6.0589	5.8563
recall@100	10.2794	7.7590	8.6926	8.5075	8.3408	7.7881	8.7896	8.9905	8.5740	7.8463	7.6053	8.2434	8.1887
ndcg@5	1.3120	1.2984	0.7801	1.2788	0.8753	0.8471	1.3027	1.0382	1.1243	1.3165	1.3125	0.9563	1.1759
ndcg@10	1.6578	1.5102	1.2275	1.6398	1.2551	1.2835	1.5710	1.4272	1.3764	1.5835	1.5538	1.3844	1.4195
ndcg@30	2.3575	2.1236	1.9236	2.2168	1.7632	1.9471	2.1405	2.0239	1.9674	2.1877	2.3538	2.0098	2.1175
ndcg@50	2.7991	2.3918	2.3269	2.5172	2.0679	2.1623	2.5461	2.3559	2.3677	2.4725	2.6340	2.3800	2.4266
ndcg@100	3.5170	2.8230	2.7995	3.0799	2.7134	2.6871	3.1225	2.9844	2.9681	2.9310	2.9541	2.8583	2.9079
Process_data_time													
seconds	291.1387	277.7499	278.3283	280.5989	279.9142	281.9208	282.1760	289.9873	290.3195	296.7917	292.3971	293.4979	292.9487
hrs	0:04:51	0:04:38	0:04:38	0:04:41	0:04:40	0:04:42	0:04:42	0:04:50	0:04:50	0:04:57	0:04:52	0:04:53	0:04:53

Y_train_dim	(4000, 7088)	Label Cardinality	2.6660	Training Samples/Label	0.5643
Y_test_dim	(1000, 7088)	Label Density	3.7613E-04		

ตาราง 4.18 : เปรียบเทียบผลข้อมูล Linear 12201-17200 ชุดทดสอบ สำหรับสัดส่วนการลดมิติ 1.0 ระหว่างแบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับ แบบจำลองตั้งต้น

Random Method	Linear 12201-17200	Testing Result											
XMLC-PAO Ratio	org	XMLC-PAO-1 : 1.0						XMLC-PAO-2 : 1.0					
Adam Learning	0.003	0.0015	0.003	0.006	0.008	0.01	0.02	0.0015	0.003	0.006	0.008	0.01	0.02
mrr	0.0264	0.0238	0.0229	0.0191	0.0226	0.0224	0.0200	0.0214	0.0227	0.0251	0.0234	0.0211	0.0194
recall@5	1.8783	1.8200	1.4761	1.0733	1.5850	1.6218	1.2964	1.0060	1.2750	1.7492	1.7243	1.7750	1.9200
recall@10	2.6817	2.5170	2.5182	1.6217	2.5350	2.1343	2.1839	1.8569	2.3293	2.8751	2.4110	2.4667	2.3492
recall@30	5.0109	4.5248	4.5609	4.1621	4.6493	4.0465	3.5750	4.0810	4.9450	4.7504	4.1204	3.9232	3.5364
recall@50	6.7852	5.5144	6.2487	5.6632	5.5509	4.8979	5.3588	5.7260	6.3815	5.8755	5.4679	4.9416	4.7694
recall@100	10.2794	8.0261	8.2346	8.1420	7.8663	7.8382	8.0363	7.9211	9.0050	8.6311	8.1676	7.0383	6.4501
ndcg@5	1.3120	1.3736	1.1117	0.8358	1.1658	1.2015	0.9502	0.8990	1.0182	1.3924	1.3206	1.1665	1.2946
ndcg@10	1.6578	1.6599	1.5321	1.0751	1.5539	1.4174	1.3109	1.2335	1.4540	1.8191	1.6113	1.4548	1.4590
ndcg@30	2.3575	2.2355	2.1546	1.8113	2.1916	1.9989	1.7407	1.8828	2.2099	2.3585	2.1395	1.9018	1.8458
ndcg@50	2.7991	2.4772	2.5407	2.1810	2.4206	2.2119	2.1564	2.2607	2.5551	2.6182	2.4624	2.1425	2.1372
ndcg@100	3.5170	2.9970	2.9452	2.6872	2.9046	2.8165	2.6960	2.7550	3.0902	3.1738	3.0180	2.5747	2.4962
Process_data_time													
seconds	291.1387	283.6918	286.4412	285.7947	287.4845	289.7898	289.4181	296.4826	298.6041	300.3685	300.7847	300.2447	301.4797
hrs	0:04:51	0:04:44	0:04:46	0:04:46	0:04:47	0:04:50	0:04:49	0:04:56	0:04:59	0:05:00	0:05:01	0:05:00	0:05:01

Y_train_dim	(4000, 7088)	Label Cardinality	2.6660	Training Samples/Label	0.5643
Y_test_dim	(1000, 7088)	Label Density	3.7613E-04		

ตาราง 4.19 : เปรียบเทียบผลข้อมูล Linear 15001-20000 ชุดทดสอบ สำหรับสัดส่วนการลดมิติ 0.5 ระหว่างแบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับ แบบจำลองตั้งต้น

Random Method	Linear 15001-20000			Testing Result									
XMLC-PAO Ratio	org	XMLC-PAO-1 : 0.5						XMLC-PAO-2 : 0.5					
Adam Learning	0.003	0.0015	0.003	0.006	0.008	0.01	0.02	0.0015	0.003	0.006	0.008	0.01	0.02
mrr	**0.0281**	0.0232	0.0226	0.0255	0.0240	0.0258	0.0212	0.0210	0.0245	0.0221	0.0240	0.0204	0.0176
recall@5	1.3793	1.2581	1.3140	1.2209	1.1559	**1.7790**	1.0517	1.0731	1.5023	1.2881	1.2850	0.9814	1.4124
recall@10	**2.7610**	2.4373	2.1657	1.8484	1.8009	2.2550	1.8000	1.5657	2.2627	1.9498	2.1445	1.2267	2.5774
recall@30	**5.4221**	4.3895	3.7533	4.2080	3.6984	3.9415	3.9825	3.7177	4.2489	4.3512	3.8693	3.0405	4.0131
recall@50	**7.0450**	5.7323	5.0155	5.6458	5.2476	5.4067	5.5834	5.1920	6.2317	5.7962	5.0624	4.7688	5.2688
recall@100	**9.7638**	8.4602	7.0399	7.3205	7.1627	9.2561	7.9592	7.6120	9.4118	7.4909	7.1229	6.8211	7.1717
ndcg@5	1.1955	1.0714	1.0958	1.1369	1.0754	**1.3613**	0.9176	0.9940	1.2067	1.0693	1.1097	0.8951	0.9161
ndcg@10	**1.7134**	1.5559	1.4664	1.3846	1.3291	1.5645	1.2601	1.1902	1.5293	1.3414	1.4725	0.9964	1.3635
ndcg@30	**2.5809**	2.1557	1.9492	2.0999	1.8894	2.0700	1.9195	1.8539	2.1215	2.0428	1.9935	1.5227	1.8255
ndcg@50	**2.9713**	2.4629	2.2528	2.4287	2.2345	2.4302	2.2899	2.2149	2.6005	2.3800	2.2871	1.9204	2.1170
ndcg@100	**3.5669**	2.9940	2.6993	2.7978	2.6600	3.1988	2.7783	2.7095	3.2616	2.7275	2.7115	2.3510	2.5018
Process_data_time													
seconds	279.5561	274.7197	**262.6380**	264.1856	264.3792	265.5783	267.0645	286.0428	274.8751	275.3273	276.4055	277.9537	277.9149
hrs	0:04:40	0:04:35	**0:04:23**	0:04:24	0:04:24	0:04:26	0:04:27	0:04:46	0:04:35	0:04:35	0:04:36	0:04:38	0:04:38

Y_train_dim (4000, 7228) **Label Cardinality** 2.6535 **Training Samples/Label** 0.5534
Y_test_dim (1000, 7228) **Label Density** 3.6711E-04

ตาราง 4.20 : เปรียบเทียบผลข้อมูล Linear 15001-20000 ชุดทดสอบ สำหรับสัดส่วนการลดมิติ 0.8 ระหว่างแบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับ แบบจำลองตั้งต้น

Random Method	Linear 15001-20000			Testing Result									
XMLC-PAO Ratio	org	XMLC-PAO-1 : 0.8						XMLC-PAO-2 : 0.8					
Adam Learning	0.003	0.0015	0.003	0.006	0.008	0.01	0.02	0.0015	0.003	0.006	0.008	0.01	0.02
mrr	**0.0281**	0.0249	0.0215	0.0211	0.0215	0.0230	0.0246	0.0235	0.0205	0.0243	0.0256	0.0211	0.0220
recall@5	1.3793	1.1233	1.1892	0.8440	1.0624	1.2925	1.3099	1.6007	1.2440	**1.7673**	1.2858	1.3390	1.1876
recall@10	**2.7610**	1.8036	1.6442	1.7758	2.2226	2.1337	2.3542	2.3499	2.3460	2.4099	1.6627	2.0823	1.8476
recall@30	**5.4221**	3.7441	3.5572	3.7143	3.9630	3.9348	4.5884	3.7342	4.6415	4.1796	4.9694	4.2171	3.3729
recall@50	**7.0450**	5.4835	4.6907	4.9747	5.5144	5.1572	6.1882	4.5554	5.8208	5.5332	7.0207	5.6073	4.3713
recall@100	**9.7638**	8.3798	6.6667	7.2901	7.4815	7.5123	7.8355	6.3073	8.1634	7.5404	9.0570	7.8433	6.7151
ndcg@5	1.1955	1.0310	0.9767	0.8045	0.9591	1.0954	1.1376	1.2879	0.9592	**1.3874**	1.1470	1.0917	1.0197
ndcg@10	**1.7134**	1.2914	1.1669	1.2038	1.4186	1.4254	1.5859	1.6002	1.4215	1.6688	1.3150	1.3811	1.2649
ndcg@30	**2.5809**	1.8721	1.7591	1.7647	1.9210	1.9157	2.2487	2.0204	2.0925	2.1857	2.2821	2.0121	1.7232
ndcg@50	**2.9713**	2.2981	2.0314	2.0777	2.3107	2.1859	2.6294	2.2176	2.3861	2.4968	2.7583	2.3487	1.9669
ndcg@100	**3.5669**	2.8826	2.4510	2.5675	2.7263	2.6752	2.9754	2.5657	2.8820	2.9540	3.2005	2.8213	2.4581
Process_data_time													
seconds	279.5561	**268.9874**	269.5131	270.9056	271.2401	272.7897	272.5271	284.5734	285.5161	287.6207	287.0519	290.1875	289.2449
hrs	0:04:40	**0:04:29**	0:04:30	0:04:31	0:04:31	0:04:33	0:04:33	0:04:45	0:04:46	0:04:48	0:04:47	0:04:50	0:04:49

Y_train_dim (4000, 7228) **Label Cardinality** 2.6535 **Training Samples/Label** 0.5534
Y_test_dim (1000, 7228) **Label Density** 3.6711E-04

ตาราง 4.21 : เปรียบเทียบผลข้อมูล Linear 15001-20000 ชุดทดสอบ สำหรับสัดส่วนการลดมิติ 1.0 ระหว่างแบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับ แบบจำลองตั้งต้น

Random Method	Linear 15001-20000	Testing Result											
XMLC-PAO Ratio	org	XMLC-PAO-1 : 1.0						XMLC-PAO-2 : 1.0					
Adam Learning	0.003	0.0015	0.003	0.006	0.008	0.01	0.02	0.0015	0.003	0.006	0.008	0.01	0.02
mrr	0.0281	0.0241	0.0257	0.0251	0.0234	0.0238	0.0243	0.0254	0.0232	0.0239	0.0203	0.0258	0.0224
recall@5	1.3793	1.0817	1.2731	1.1182	1.1212	1.8442	1.3450	1.1543	1.3814	1.3203	0.9674	1.4120	1.3383
recall@10	2.7610	1.8544	2.5542	1.9544	1.8330	2.4178	2.0268	1.9076	2.4183	2.2003	1.5470	1.9497	1.9117
recall@30	5.4221	3.9734	4.9338	3.9643	4.0802	4.3516	4.0440	4.1190	3.9143	4.1221	3.0747	4.2748	3.8559
recall@50	7.0450	5.5969	6.0228	5.5986	5.1382	5.6124	5.2046	5.3357	5.1223	5.9621	4.4924	5.8516	5.8376
recall@100	9.7638	7.8890	7.7843	7.8881	6.7123	7.4880	8.3318	7.3056	7.9085	8.3066	7.4315	7.4798	8.3988
ndcg@5	1.1955	1.0207	1.1187	1.0401	0.9721	1.3886	1.1219	1.1234	1.0994	1.1499	0.8855	1.2197	0.9909
ndcg@10	1.7134	1.3496	1.6395	1.3968	1.2715	1.6521	1.4100	1.4139	1.5495	1.4966	1.1272	1.4555	1.2179
ndcg@30	2.5809	2.0216	2.3859	2.0280	1.9351	2.2496	1.9927	2.0312	2.0194	2.0617	1.5790	2.1759	1.8434
ndcg@50	2.9713	2.4016	2.6612	2.4244	2.2069	2.5406	2.2847	2.3084	2.2978	2.4973	1.9358	2.5493	2.3096
ndcg@100	3.5669	2.8425	3.0199	2.9081	2.5446	2.9418	2.9197	2.7140	2.9015	3.0006	2.5596	2.8941	2.8378
Process_data_time													
seconds	279.5561	274.9714	277.1300	277.0900	277.7818	279.8261	279.3560	294.0639	294.8051	294.5351	296.0705	298.0651	298.1227
hrs	0:04:40	0:04:35	0:04:37	0:04:37	0:04:38	0:04:40	0:04:39	0:04:54	0:04:55	0:04:55	0:04:56	0:04:58	0:04:58

Y_train_dim	(4000, 7228)	Label Cardinality	2.6535	Training Samples/Label	0.5534
Y_test_dim	(1000, 7228)	Label Density	3.6711E-04		

ตาราง 4.22 : เปรียบเทียบผลข้อมูล Linear 20001-25000 ชุดทดสอบ สำหรับสัดส่วนการลดมิติ 0.5 ระหว่างแบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับ แบบจำลองตั้งต้น

Random Method	Linear 20001-25000	Testing Result											
XMLC-PAO Ratio	org	XMLC-PAO-1 : 0.5						XMLC-PAO-2 : 0.5					
Adam Learning	0.003	0.0015	0.003	0.006	0.008	0.01	0.02	0.0015	0.003	0.006	0.008	0.01	0.02
mrr	0.0192	0.0168	0.0180	0.0181	0.0162	0.0163	0.0177	0.0177	0.0185	0.0162	0.0149	0.0143	0.0149
recall@5	1.1059	1.1531	1.2825	1.6964	0.9825	1.4673	1.2942	1.0248	1.3692	1.0309	0.9386	1.0284	0.9109
recall@10	2.5559	1.9225	2.1859	2.3114	1.8942	2.1539	1.8725	1.8309	2.5918	2.1225	2.3600	1.8492	1.9059
recall@30	5.3993	3.5284	3.9327	4.6385	3.9750	3.9291	4.2942	4.7201	4.4642	4.4377	4.3735	3.8109	3.6215
recall@50	7.1867	5.2103	5.1161	6.0804	5.5183	5.2617	5.5753	6.1442	5.8151	5.8804	5.4435	5.0463	4.8632
recall@100	10.0068	7.3001	6.9719	8.6488	7.2942	7.5362	8.2741	7.9612	8.3795	8.0720	6.7895	7.4946	6.3992
ndcg@5	0.7972	0.8238	0.9526	1.0567	0.7101	0.9653	0.9671	0.7960	0.9301	0.7127	0.6646	0.7599	0.6605
ndcg@10	1.3337	1.0880	1.2855	1.2981	1.0508	1.2385	1.1888	1.1170	1.3920	1.1032	1.1791	1.0734	1.0531
ndcg@30	2.1367	1.5316	1.7615	1.9611	1.6592	1.7627	1.8300	1.8787	1.9192	1.7455	1.7453	1.6094	1.5569
ndcg@50	2.5313	1.9054	2.0355	2.3002	2.0119	2.0760	2.1341	2.1864	2.2456	2.0645	1.9971	1.8993	1.8204
ndcg@100	3.0859	2.3220	2.4153	2.8212	2.3801	2.5336	2.6676	2.5465	2.7562	2.5203	2.2726	2.4000	2.1299
Process_data_time													
seconds	291.5006	271.8910	273.4393	274.2250	274.8275	278.9125	279.9659	288.3216	277.1843	277.6070	279.3432	279.0132	280.8723
hrs	0:04:52	0:04:32	0:04:33	0:04:34	0:04:35	0:04:39	0:04:40	0:04:48	0:04:37	0:04:38	0:04:39	0:04:39	0:04:41

Y_train_dim	(4000, 6931)	Label Cardinality	2.4700	Training Samples/Label	0.5771
Y_test_dim	(1000, 6931)	Label Density	3.5637E-04		

ตาราง 4.23 : เปรียบเทียบผลข้อมูล Linear 20001-25000 ชุดทดสอบ สำหรับสัดส่วนการลดมิติ 0.8 ระหว่างแบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับ แบบจำลองตั้งต้น

Random Method	Linear 20001-25000	Testing Result											
XMLC-PAO Ratio	org	XMLC-PAO-1 : 0.8						XMLC-PAO-2 : 0.8					
Adam Learning	0.003	0.0015	0.003	0.006	0.008	0.01	0.02	0.0015	0.003	0.006	0.008	0.01	0.02
mrr	0.0192	0.0182	0.0155	0.0187	0.0195	0.0132	0.0149	0.0162	0.0146	0.0143	0.0148	0.0142	0.0186
recall@5	1.1059	1.4198	0.7864	1.5364	1.7698	0.5531	0.9442	1.1114	1.0948	0.8164	0.8475	0.9725	1.4531
recall@10	2.5559	2.0981	1.5064	2.2648	2.8656	1.2198	1.5842	1.9064	1.3114	1.4835	1.4850	1.9267	2.1189
recall@30	5.3993	3.3956	3.2677	4.4874	4.5735	2.8390	3.5156	3.5294	2.9923	3.2352	3.9943	3.8091	4.1274
recall@50	7.1867	4.4283	5.1177	5.7346	5.6833	3.8789	4.2502	4.8935	4.3339	4.3185	5.4760	4.3541	5.3515
recall@100	10.0068	6.6791	7.9371	7.4939	7.1640	6.6475	6.3206	7.3265	6.9567	6.7614	7.0068	6.3123	7.3623
ndcg@5	0.7972	1.0442	0.6385	1.1382	1.2014	0.4657	0.6728	0.7609	0.7967	0.5845	0.6415	0.6829	1.0000
ndcg@10	1.3337	1.2975	0.9347	1.4056	1.6275	0.7085	0.9100	1.0631	0.8830	0.8510	0.9024	1.0412	1.2551
ndcg@30	2.1367	1.6783	1.4149	2.0410	2.1244	1.1868	1.4635	1.5301	1.3717	1.3570	1.6168	1.5842	1.8143
ndcg@50	2.5313	1.9123	1.8100	2.3109	2.3850	1.4463	1.6590	1.8191	1.6576	1.6031	1.9341	1.7191	2.0824
ndcg@100	3.0859	2.3509	2.3868	2.6916	2.7049	2.0230	2.0452	2.2924	2.1806	2.0987	2.2461	2.0999	2.4919
Process_data_time													
seconds	291.5006	280.5030	280.8394	281.2258	282.3380	282.8808	285.6473	285.2144	288.2474	286.3446	289.1967	288.7589	290.9288
hrs	0:04:52	0:04:41	0:04:41	0:04:41	0:04:42	0:04:43	0:04:46	0:04:45	0:04:48	0:04:46	0:04:49	0:04:49	0:04:51

Y_train_dim	(4000, 6931)	Label Cardinality	2.4700	Training Samples/Label	0.5771
Y_test_dim	(1000, 6931)	Label Density	3.5637E-04		

ตาราง 4.24 : เปรียบเทียบผลข้อมูล Linear 20001-25000 ชุดทดสอบ สำหรับสัดส่วนการลดมิติ 1.0 ระหว่างแบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับ แบบจำลองตั้งต้น

Random Method	Linear 20001-25000	Testing Result											
XMLC-PAO Ratio	org	XMLC-PAO-1 : 1.0						XMLC-PAO-2 : 1.0					
Adam Learning	0.003	0.0015	0.003	0.006	0.008	0.01	0.02	0.0015	0.003	0.006	0.008	0.01	0.02
mrr	0.0192	0.0153	0.0157	0.0168	0.0149	0.0149	0.0161	0.0179	0.0171	0.0185	0.0155	0.0133	0.0149
recall@5	1.1059	1.0314	1.1892	1.5048	1.0031	0.5998	1.4064	1.2114	0.8509	1.2948	0.8817	1.1025	0.9475
recall@10	2.5559	2.1814	1.7225	2.4114	2.0102	1.0731	2.2009	2.4309	2.3909	2.2981	1.8517	1.7525	1.9309
recall@30	5.3993	3.4598	4.4734	3.7417	4.1662	4.3784	3.2443	4.4551	3.7078	4.0877	3.8601	3.6992	3.8675
recall@50	7.1867	4.7875	5.9762	4.6050	5.7104	5.6252	4.3927	5.7353	5.3786	5.4671	5.4717	4.9246	5.2575
recall@100	10.0068	6.9743	7.4999	7.1941	7.7712	7.8977	6.2579	8.3670	6.9812	7.0972	7.5337	7.2421	7.9424
ndcg@5	0.7972	0.7544	0.8348	1.0835	0.6654	0.5617	0.9918	0.8311	0.7068	0.9867	0.6726	0.6408	0.6906
ndcg@10	1.3337	1.1628	1.0538	1.4505	1.0523	0.7478	1.2951	1.2645	1.2819	1.3260	1.0231	0.8764	1.0673
ndcg@30	2.1367	1.5255	1.7961	1.8722	1.6370	1.6986	1.5875	1.8490	1.6784	1.8444	1.5856	1.4680	1.5725
ndcg@50	2.5313	1.8308	2.1226	2.0758	1.9981	1.9975	1.8329	2.1527	2.0224	2.1801	1.9433	1.7494	1.8885
ndcg@100	3.0859	2.2753	2.4416	2.5958	2.4061	2.4264	2.2253	2.6575	2.3567	2.5145	2.3586	2.2242	2.4462
Process_data_time													
seconds	291.5006	286.2104	288.3949	289.3650	290.0340	290.7418	291.1257	293.4787	295.6047	295.1565	297.2512	296.8167	299.7163
hrs	0:04:52	0:04:46	0:04:48	0:04:49	0:04:50	0:04:51	0:04:51	0:04:53	0:04:56	0:04:55	0:04:57	0:04:57	0:05:00

Y_train_dim (4000, 6931) **Label Cardinality** 2.4700 **Training Samples/Label** 0.5771

Y_test_dim (1000, 6931) **Label Density** 3.5637E-04

ตาราง 4.25 : เปรียบเทียบผลข้อมูล Linear 25001-27932 (2933 ตัวอย่าง) ชุดทดสอบ สำหรับสัดส่วนการลดมิติ 0.5 ระหว่างแบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับแบบจำลองตั้งต้น

Random Method	Linear 25001-27932	Testing Result											
XMLC-PAO Ratio	org	XMLC-PAO-1 : 0.5						XMLC-PAO-2 : 0.5					
Adam Learning	0.003	0.0015	0.003	0.006	0.008	0.01	0.02	0.0015	0.003	0.006	0.008	0.01	0.02
mrr	0.0226	0.0190	0.0213	0.0191	0.0214	0.0202	0.0153	0.0182	0.0206	**0.0248**	0.0219	0.0203	0.0161
recall@5	1.8881	1.1499	1.9449	1.3913	1.4906	1.7036	1.0363	1.6184	1.2635	**2.0301**	1.2777	1.2209	1.1357
recall@10	2.0585	1.8171	**2.4134**	1.8597	1.8455	2.1863	1.5474	1.8739	1.8228	2.2572	1.9165	1.9875	1.2635
recall@30	3.6400	3.4072	3.6684	3.1958	3.0437	**3.9949**	2.4276	3.2567	3.6513	3.8063	3.4270	3.8189	2.8778
recall@50	5.1732	4.3867	4.1936	4.2322	4.1936	5.0272	3.5619	4.3697	4.8296	**5.1833**	5.0170	4.8284	3.9908
recall@100	**8.2185**	6.5275	5.9455	5.1521	6.3714	7.4605	6.2791	6.2209	6.5758	6.5178	7.2007	6.6768	6.1317
ndcg@5	1.3323	0.9758	1.3232	1.1132	1.1505	1.1694	0.7970	1.1515	1.1037	**1.5583**	1.0681	0.9088	0.9192
ndcg@10	1.4079	1.2686	1.5261	1.2870	1.2977	1.3577	1.0091	1.2436	1.3193	**1.6582**	1.3108	1.1986	0.9820
ndcg@30	1.9131	1.6966	1.9109	1.6553	1.6177	1.8372	1.2660	1.6348	1.8098	**2.0996**	1.7327	1.7404	1.4506
ndcg@50	2.2467	1.8997	2.0317	1.9035	1.8955	2.0762	1.5445	1.8942	2.0706	**2.4052**	2.0773	1.9689	1.7001
ndcg@100	**2.8264**	2.3192	2.3734	2.0878	2.3129	2.5523	2.0726	2.2439	2.4009	2.6774	2.5285	2.3314	2.1545
Process_data_time													
seconds	170.1603	**155.9716**	156.3979	156.8777	157.6429	158.4585	159.3564	159.3495	160.2767	160.7963	161.2509	162.4565	162.9972
hrs	0:02:50	**0:02:36**	0:02:36	0:02:37	0:02:38	0:02:38	0:02:39	0:02:39	0:02:40	0:02:41	0:02:41	0:02:42	0:02:43

Y_train_dim	(2346, 4792)	Label Cardinality	2.4224	Training Samples/Label	0.4896
Y_test_dim	(587, 4792)	Label Density	5.0551E-04		

ตาราง 4.26 : เปรียบเทียบผลข้อมูล Linear 25001-27932 (2933 ตัวอย่าง) ชุดทดสอบ สำหรับสัดส่วนการลดมิติ 0.8 ระหว่างแบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับแบบจำลองตั้งต้น

Random Method	Linear 25001-27932	Testing Result											
XMLC-PAO Ratio	org	XMLC-PAO-1 : 0.8						XMLC-PAO-2 : 0.8					
Adam Learning	0.003	0.0015	0.003	0.006	0.008	0.01	0.02	0.0015	0.003	0.006	0.008	0.01	0.02
mrr	**0.0226**	0.0201	0.0173	0.0176	0.0193	0.0142	0.0191	0.0179	0.0202	0.0166	0.0188	0.0165	0.0185
recall@5	**1.8881**	1.6042	1.4055	1.1925	1.1925	1.3771	1.3771	1.1215	1.4196	1.4338	1.4764	1.2493	1.3629
recall@10	2.0585	2.1153	**2.3850**	1.4622	1.4906	1.6326	1.7178	1.4338	1.9023	1.9449	2.1579	1.3487	1.6468
recall@30	3.6400	3.5640	3.4923	1.8881	2.4560	2.5554	**3.7266**	2.0074	3.5633	3.5974	3.2709	2.9602	3.1674
recall@50	**5.1732**	4.2653	4.5627	3.0636	2.8677	3.1374	4.7232	3.3333	4.5287	4.6479	5.1363	4.3940	4.0249
recall@100	**8.2185**	6.1250	7.1806	6.3174	4.7473	4.9319	7.7953	6.2126	6.8910	6.5999	6.3629	6.0749	5.3977
ndcg@5	**1.3323**	1.1136	0.9510	0.9588	1.0626	0.9228	1.1185	0.8591	1.0705	0.9839	1.0703	0.8432	1.0313
ndcg@10	**1.4079**	1.2997	1.2889	1.0702	1.1784	1.0127	1.2266	1.0087	1.2702	1.1757	1.3221	0.8909	1.1343
ndcg@30	**1.9131**	1.7307	1.6203	1.1927	1.4913	1.2820	1.7637	1.1946	1.7369	1.6573	1.6299	1.3959	1.6230
ndcg@50	**2.2467**	1.8889	1.8708	1.4635	1.5843	1.4381	1.9925	1.5027	1.9637	1.9124	2.0637	1.6832	1.8146
ndcg@100	**2.8264**	2.2491	2.3735	2.0752	1.9464	1.7810	2.5605	2.0411	2.4418	2.3185	2.3038	2.0070	2.1046
Process_data_time													
seconds	170.1603	**160.1624**	160.9981	162.4114	162.5840	163.1437	163.8967	164.2075	165.6152	165.8609	166.6519	167.5537	169.1035
hrs	0:02:50	**0:02:40**	0:02:41	0:02:42	0:02:43	0:02:43	0:02:44	0:02:44	0:02:46	0:02:46	0:02:47	0:02:48	0:02:49

Y_train_dim	(2346, 4792)	Label Cardinality	2.4224	Training Samples/Label	0.4896
Y_test_dim	(587, 4792)	Label Density	5.0551E-04		

ตาราง 4.27 : เปรียบเทียบผลข้อมูล Linear 25001-27932 (2933 ตัวอย่าง) ชุดทดสอบ สำหรับสัดส่วนการลดมิติ 1.0 ระหว่างแบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับแบบจำลองตั้งต้น

Random Method	Linear 25001-27932	Testing Result											
XMLC-PAO Ratio	org	XMLC-PAO-1 : 1.0						XMLC-PAO-2 : 1.0					
Adam Learning	0.003	0.0015	0.003	0.006	0.008	0.01	0.02	0.0015	0.003	0.006	0.008	0.01	0.02
mrr	0.0226	0.0177	0.0181	0.0186	0.0198	0.0162	0.0159	0.0159	0.0201	0.0176	0.0130	0.0223	0.0194
recall@5	1.8881	0.9654	1.2777	1.2777	1.5474	1.1215	1.3913	0.8234	1.4906	1.4196	0.6672	1.2919	0.9512
recall@10	2.0585	1.5048	1.8455	1.5474	1.9080	1.4055	1.5190	1.5190	1.5389	2.9245	0.8802	1.7036	1.6711
recall@30	3.6400	2.8393	3.1232	3.0863	2.9443	2.7683	3.0239	2.9302	3.3889	4.2362	2.2771	2.9160	3.1248
recall@50	5.1732	3.9182	3.6343	3.6315	4.3668	4.6053	4.0645	3.5945	4.3032	4.9602	3.9949	3.3135	4.2606
recall@100	8.2185	5.3638	5.5622	5.2271	7.2535	6.5133	5.3194	5.5253	5.3783	6.5843	5.8745	5.6956	6.0976
ndcg@5	1.3323	0.8282	0.9739	0.9522	1.0933	0.9052	1.0047	0.7544	1.0704	0.9836	0.5492	1.2306	0.9249
ndcg@10	1.4079	1.0359	1.1901	1.0603	1.2358	1.0269	1.0699	0.9995	1.0976	1.5379	0.6301	1.4018	1.1916
ndcg@30	1.9131	1.3821	1.5281	1.5474	1.5684	1.4009	1.4751	1.4157	1.6536	1.9531	1.0264	1.7498	1.6237
ndcg@50	2.2467	1.6232	1.6619	1.6747	1.8816	1.8037	1.7277	1.5790	1.8501	2.1231	1.4130	1.8435	1.9022
ndcg@100	2.8264	1.9198	2.0396	1.9660	2.4368	2.1738	2.0024	1.9736	2.0688	2.4505	1.7791	2.3230	2.2559
Process_data_time													
seconds	170.1603	165.1016	166.5484	166.4967	167.2005	169.3134	168.7673	169.6006	170.7693	171.4299	172.7543	173.1211	175.9085
hrs	0:02:50	0:02:45	0:02:47	0:02:46	0:02:47	0:02:49	0:02:49	0:02:50	0:02:51	0:02:51	0:02:53	0:02:53	0:02:56

Y_train_dim	(2346, 4792)	Label Cardinality	2.4224	Training Samples/Label	0.4896
Y_test_dim	(587, 4792)	Label Density	5.0551E-04		

ตาราง 4.28 : เปรียบเทียบผลข้อมูล Random seed-42 ชุดทดสอบ สำหรับสัดส่วนการลดมิติ 0.5 ระหว่างแบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับ แบบจำลองดั้งต้น

Random Method	Random seed-42	Testing Result											
XMLC-PAO Ratio	org	XMLC-PAO-1 : 0.5						XMLC-PAO-2 : 0.5					
Adam Learning	0.003	0.0015	0.003	0.006	0.008	0.01	0.02	0.0015	0.003	0.006	0.008	0.01	0.02
mrr	0.0280	0.0172	0.0212	0.0212	0.0188	0.0171	0.0250	0.0219	0.0218	0.0231	0.0237	0.0251	0.0243
recall@5	0.8847	0.5371	0.6972	0.9484	1.0129	0.7833	0.8521	0.7760	0.9029	1.0532	0.9932	0.9591	1.0456
recall@10	2.1019	1.0380	1.7124	1.3890	1.4525	0.9954	1.8012	1.1787	1.2061	1.7141	1.4262	1.3844	1.6036
recall@30	4.1918	2.2252	3.1539	2.4311	2.2191	2.2065	2.8709	3.7733	2.2630	3.3447	3.0969	3.0422	3.0111
recall@50	5.2065	2.8604	4.2473	2.9227	2.9883	2.7342	3.8719	4.7667	3.2487	4.5046	3.8059	4.1652	4.3579
recall@100	7.0344	4.6084	6.4279	6.0290	4.2547	4.8668	5.6349	6.2591	4.9482	6.4263	5.3114	5.5874	5.8861
ndcg@5	1.0024	0.5951	0.8122	0.8464	0.8374	0.7048	0.9548	0.8079	0.8876	0.9550	1.0175	0.9897	1.0436
ndcg@10	1.4981	0.8144	1.1727	1.0476	1.0224	0.8054	1.3428	1.0116	1.0255	1.2401	1.1949	1.1965	1.2686
ndcg@30	2.1566	1.1546	1.5805	1.3920	1.2673	1.1868	1.7150	1.7729	1.3416	1.7211	1.7084	1.7200	1.7009
ndcg@50	2.4134	1.3210	1.8295	1.5530	1.4518	1.3230	1.9739	1.9962	1.5988	1.9768	1.9046	1.9806	2.0257
ndcg@100	2.8011	1.7044	2.3026	2.2240	1.7213	1.7492	2.3380	2.3339	1.9634	2.4062	2.2561	2.3030	2.3543
Process_data_time													
seconds	269.3646	281.5127	268.7470	268.2193	269.3863	272.0557	272.7078	304.1104	305.9943	302.6955	304.8159	303.5382	305.3572
hrs	0:04:29	0:04:35	0:04:35	0:04:35	0:04:36	0:04:36	0:04:38	0:05:17	0:05:18	0:05:19	0:05:32	0:05:24	0:05:27

Y_train_dim (4000, 8468) Label Cardinality 3.1473 Training Samples/Label 0.4724
Y_test_dim (1000, 8468) Label Density 3.7166E-04

ตาราง 4.29 : เปรียบเทียบผลข้อมูล Random seed-42 ชุดทดสอบ สำหรับสัดส่วนการลดมิติ 0.8 ระหว่างแบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับ แบบจำลองดั้งต้น

Random Method	Random seed-42	Testing Result											
XMLC-PAO Ratio	org	XMLC-PAO-1 : 0.8						XMLC-PAO-2 : 0.8					
Adam Learning	0.003	0.0015	0.003	0.006	0.008	0.01	0.02	0.0015	0.003	0.006	0.008	0.01	0.02
mrr	0.0280	0.0254	0.0162	0.0252	0.0248	0.0244	0.0228	0.0257	0.0252	0.0213	0.0262	0.0112	0.0261
recall@5	0.8847	1.0715	0.6216	1.1126	0.9551	1.0551	0.9919	1.1715	1.3065	0.7891	1.3594	0.4240	0.8236
recall@10	2.1019	1.6968	1.1459	1.7923	1.6438	1.5235	1.3017	1.8701	1.6586	1.1950	2.4062	0.4690	1.5272
recall@30	4.1918	3.3098	2.0650	3.2633	2.4461	3.3139	1.6593	3.5595	3.3594	2.5014	4.0830	0.6569	3.1756
recall@50	5.2065	4.0165	2.3922	3.9767	4.0367	4.5446	2.3154	4.3594	4.3844	3.3547	5.1050	0.8552	3.9239
recall@100	7.0344	6.4605	5.3907	5.9668	6.2722	5.9461	4.3755	5.7816	6.2259	5.5164	7.3357	3.4855	5.9226
ndcg@5	1.0024	1.0981	0.6266	1.1009	1.0277	1.0137	1.0350	1.0720	1.1522	0.8643	1.1725	0.4400	0.9557
ndcg@10	1.4981	1.3427	0.8331	1.3882	1.2995	1.2117	1.1772	1.3790	1.3252	1.0441	1.5757	0.4632	1.2378
ndcg@30	2.1566	1.8534	1.0949	1.8334	1.5779	1.7534	1.3119	1.9076	1.8745	1.4335	2.0618	0.5311	1.7755
ndcg@50	2.4134	2.0152	1.1718	2.0130	1.9483	2.0475	1.4757	2.1044	2.1318	1.6450	2.3159	0.5759	1.9857
ndcg@100	2.8011	2.5490	1.8117	2.4555	2.4215	2.3324	1.8891	2.4136	2.5657	2.0666	2.7962	1.1058	2.3939
Process_data_time													
seconds	269.3646	275.3721	275.0418	274.5494	276.1146	275.5278	278.2145	316.9397	317.6689	319.1348	331.6014	323.7464	326.9723
hrs	0:04:29	0:04:35	0:04:35	0:04:35	0:04:36	0:04:36	0:04:38	0:05:17	0:05:18	0:05:19	0:05:32	0:05:24	0:05:27

Y_train_dim (4000, 8468) Label Cardinality 3.1473 Training Samples/Label 0.4724
Y_test_dim (1000, 8468) Label Density 3.7166E-04

ตาราง 4.30 : เปรียบเทียบผลข้อมูล Random seed-42 ชุดทดสอบ สำหรับสัดส่วนการลดมิติ 1.0 ระหว่างแบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับ แบบจำลองดั้งต้น

Random Method	Random seed-42	Testing Result											
XMLC-PAO Ratio	org	XMLC-PAO-1 : 1.0						XMLC-PAO-2 : 1.0					
Adam Learning	0.003	0.0015	0.003	0.006	0.008	0.01	0.02	0.0015	0.003	0.006	0.008	0.01	0.02
mrr	0.0280	0.0188	0.0229	0.0195	0.0259	0.0200	0.0237	0.0145	0.0219	0.0212	0.0207	0.0181	0.0247
recall@5	0.8847	0.8593	0.8086	1.0334	1.5625	1.1359	1.0722	0.6983	0.9107	0.9723	0.9042	0.6791	0.9991
recall@10	2.1019	1.3488	1.4392	1.6446	2.0424	1.6404	2.0626	0.9456	1.3033	1.4847	1.4221	0.8797	1.6096
recall@30	4.1918	3.6279	2.8138	3.0789	3.1299	2.4533	4.3639	2.1082	2.7674	2.6882	2.0462	2.1658	2.6054
recall@50	5.2065	4.5754	3.9769	3.8663	3.9905	3.2340	5.2163	2.3869	3.7919	3.8906	2.4724	2.7736	3.6852
recall@100	7.0344	6.4414	6.4767	5.3170	6.4729	5.7429	7.1766	4.6566	4.9347	5.4500	5.5279	5.2317	5.3005
ndcg@5	1.0024	0.7195	0.8824	0.8600	1.3246	0.9818	0.9859	0.6538	0.9400	0.9504	0.9111	0.7394	1.0582
ndcg@10	1.4981	0.9451	1.1612	1.0964	1.5364	1.1932	1.3809	0.7553	1.0855	1.1926	1.1224	0.8307	1.3148
ndcg@30	2.1566	1.6135	1.5518	1.4947	1.8895	1.4337	2.0483	1.1414	1.4981	1.5632	1.3310	1.2271	1.6122
ndcg@50	2.4134	1.8532	1.8172	1.6873	2.1049	1.6243	2.2649	1.2167	1.7323	1.8394	1.4404	1.3928	1.8745
ndcg@100	2.8011	2.2682	2.3164	1.9930	2.6021	2.1469	2.6833	1.6886	2.0058	2.1638	2.0872	1.9192	2.2420
Process_data_time													
seconds	269.3646	279.5240	281.1356	279.8193	280.3088	282.1792	283.5612	312.2808	300.3509	301.8863	303.5976	302.4060	304.4920
hrs	0:04:29	0:04:35	0:04:35	0:04:35	0:04:36	0:04:36	0:04:38	0:05:17	0:05:18	0:05:19	0:05:32	0:05:24	0:05:27

Y_train_dim	(4000, 8468)	Label Cardinality	3.1473	Training Samples/Label	0.4724
Y_test_dim	(1000, 8468)	Label Density	3.7166E-04		

ตาราง 4.31 : เปรียบเทียบผลข้อมูล Random seed-101 ชุดทดสอบ สำหรับสัดส่วนการลดมิติ 0.5 ระหว่างแบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับ แบบจำลองตั้งต้น

Random Method	Random seed-101	Testing Result											
XMLC-PAO Ratio	org	XMLC-PAO-1 : 0.5						XMLC-PAO-2 : 0.5					
Adam Learning	0.003	0.0015	0.003	0.006	0.008	0.01	0.02	0.0015	0.003	0.006	0.008	0.01	0.02
mrr	0.0230	0.0190	0.0183	0.0147	0.0169	0.0177	0.0207	0.0182	0.0201	0.0204	0.0193	0.0201	0.0198
recall@5	1.1415	0.7755	0.5185	0.7381	0.7973	0.6753	1.0691	0.6719	1.1005	1.0239	0.9848	0.8890	0.8655
recall@10	1.5448	1.5308	1.1851	1.1452	1.2921	1.1266	1.3779	0.9380	1.4011	1.8010	1.3721	1.5049	1.9641
recall@30	3.8999	3.6098	3.9260	2.9294	2.9773	2.9531	3.2679	2.6781	2.9378	3.8852	2.8235	3.1340	3.4562
recall@50	5.7540	4.7774	4.5050	4.2304	3.7874	4.1134	4.7523	3.4662	4.4187	5.4078	3.8258	4.1844	4.2123
recall@100	9.6780	6.3048	6.5779	5.9670	5.7847	6.5553	6.2007	5.3074	6.5527	7.7512	6.1796	6.1690	5.8951
ndcg@5	0.9300	0.7257	0.5824	0.5760	0.6815	0.6163	0.9173	0.6547	0.8889	0.8619	0.8249	0.8394	0.7629
ndcg@10	1.1072	1.0119	0.8360	0.7138	0.8825	0.8152	1.0511	0.7870	1.0336	1.1532	0.9986	1.0600	1.1797
ndcg@30	1.8288	1.5963	1.5713	1.2857	1.3895	1.3570	1.6186	1.2829	1.4825	1.7638	1.4246	1.5152	1.6329
ndcg@50	2.2983	1.8990	1.7236	1.6054	1.5826	1.6449	1.9764	1.4902	1.8402	2.1251	1.6712	1.7780	1.8482
ndcg@100	3.0801	2.2084	2.1766	2.0101	2.0238	2.1681	2.2812	1.8839	2.2619	2.6205	2.2294	2.1969	2.2615
Process_data_time													
seconds	284.6269	279.2733	266.3802	267.2192	268.2536	269.5964	270.6657	280.7266	276.4966	277.9987	278.8824	279.9648	279.7141
hrs	0:04:45	0:04:33	0:04:33	0:04:35	0:04:36	0:04:37	0:04:38	0:04:49	0:04:50	0:04:52	0:04:52	0:04:52	0:04:52

Y_train_dim (4000, 8422) Label Cardinality 2.9568 Training Samples/Label 0.4749
Y_test_dim (1000, 8422) Label Density 3.5107E-04

ตาราง 4.32 : เปรียบเทียบผลข้อมูล Random seed-101 ชุดทดสอบ สำหรับสัดส่วนการลดมิติ 0.8 ระหว่างแบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับ แบบจำลองตั้งต้น

Random Method	Random seed-101	Testing Result											
XMLC-PAO Ratio	org	XMLC-PAO-1 : 0.8						XMLC-PAO-2 : 0.8					
Adam Learning	0.003	0.0015	0.003	0.006	0.008	0.01	0.02	0.0015	0.003	0.006	0.008	0.01	0.02
mrr	0.0230	0.0204	0.0187	0.0190	0.0189	0.0201	0.0183	0.0197	0.0191	0.0190	0.0215	0.0174	0.0185
recall@5	1.1415	0.8854	0.8705	0.9542	0.8223	0.8696	0.8963	0.7373	1.0715	0.7448	1.0248	0.8881	0.8750
recall@10	1.5448	1.4454	1.3310	1.2825	1.0675	1.4831	1.1426	1.6106	1.3248	1.3319	1.8685	1.2381	1.4365
recall@30	3.8999	2.9299	3.2755	2.5174	3.3849	3.3606	2.4706	3.9798	2.9392	3.0953	3.5552	1.8799	2.2803
recall@50	5.7540	4.0191	4.3735	3.5478	5.1004	4.1302	3.5579	4.7804	4.3978	4.0832	4.5089	2.6591	2.7668
recall@100	9.6780	6.4108	6.6359	6.1237	7.0790	6.2337	6.2145	7.0679	6.4143	6.4537	6.7812	5.0334	4.9207
ndcg@5	0.9300	0.8070	0.7994	0.8111	0.7467	0.8038	0.7928	0.7421	0.8653	0.7224	0.8935	0.7980	0.7973
ndcg@10	1.1072	1.0442	0.9611	0.9719	0.8539	1.0502	0.9192	1.0529	0.9783	0.9409	1.2349	0.9491	1.0208
ndcg@30	1.8288	1.4868	1.5119	1.3317	1.5473	1.6402	1.3005	1.7032	1.4668	1.4516	1.7206	1.1327	1.3136
ndcg@50	2.2983	1.7689	1.8085	1.5718	1.9480	1.8361	1.5523	1.9048	1.8073	1.7160	1.9542	1.3078	1.4542
ndcg@100	3.0801	2.3225	2.2818	2.1020	2.3862	2.2785	2.1062	2.3908	2.2535	2.2572	2.4199	1.8100	1.9224
Process_data_time													
seconds	284.6269	273.1241	273.4103	275.0971	275.9106	277.1411	277.6546	289.4682	290.4975	291.7195	292.1839	291.7422	292.0908
hrs	0:04:45	0:04:33	0:04:33	0:04:35	0:04:36	0:04:37	0:04:38	0:04:49	0:04:50	0:04:52	0:04:52	0:04:52	0:04:52

Y_train_dim (4000, 8422) Label Cardinality 2.9568 Training Samples/Label 0.4749
Y_test_dim (1000, 8422) Label Density 3.5107E-04

ตาราง 4.33 : เปรียบเทียบผลข้อมูล Random seed-101 ชุดทดสอบ สำหรับสัดส่วนการลดมิติ 1.0 ระหว่างแบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับ แบบจำลองตั้งต้น

Random Method	Random seed-101	Testing Result											
XMLC-PAO Ratio	org	XMLC-PAO-1 : 1.0						XMLC-PAO-2 : 1.0					
Adam Learning	0.003	0.0015	0.003	0.006	0.008	0.01	0.02	0.0015	0.003	0.006	0.008	0.01	0.02
mrr	0.0230	0.0193	0.0188	0.0196	0.0196	0.0207	0.0198	0.0204	0.0201	0.0184	0.0210	0.0177	0.0172
recall@5	1.1415	0.9922	1.0923	1.0564	1.1045	0.9628	0.8016	0.7934	1.0190	0.9153	1.1539	0.8602	1.0388
recall@10	1.5448	1.5745	1.3523	1.5656	1.6982	1.3540	1.0649	1.6908	1.5003	1.3993	1.7986	1.3605	1.1611
recall@30	3.8999	3.8231	2.2178	3.3100	3.4446	2.7449	3.4344	3.0182	2.7877	3.1255	3.1152	3.1524	2.7684
recall@50	5.7540	4.8129	3.3734	4.3450	4.4914	3.9466	4.7926	4.2459	3.7493	4.3495	4.0733	4.0619	3.5565
recall@100	9.6780	6.6171	5.5523	6.2475	6.6879	5.6328	7.2923	6.5279	6.2919	7.0933	6.1986	6.0697	6.1417
ndcg@5	0.9300	0.8222	0.8968	0.8573	0.8805	0.8954	0.7806	0.7786	0.8638	0.8122	0.9782	0.7185	0.7971
ndcg@10	1.1072	1.0456	1.0011	1.0852	1.1077	1.0650	0.8890	1.1251	1.0729	1.0040	1.2508	0.9181	0.8628
ndcg@30	1.8288	1.6666	1.2697	1.6232	1.6522	1.5113	1.5480	1.5489	1.4809	1.4978	1.6464	1.4231	1.3373
ndcg@50	2.2983	1.9088	1.5496	1.8815	1.9086	1.8104	1.8896	1.8534	1.7144	1.7818	1.8877	1.6548	1.5317
ndcg@100	3.0801	2.2839	2.0276	2.2663	2.3733	2.2152	2.3831	2.3281	2.2334	2.3552	2.3659	2.0948	2.0989
Process_data_time													
seconds	284.6269	280.6662	282.1414	281.6628	282.0380	284.5839	283.7131	288.1532	277.1801	277.9502	279.2389	279.0803	278.9957
hrs	0:04:45	0:04:33	0:04:33	0:04:35	0:04:36	0:04:37	0:04:38	0:04:49	0:04:50	0:04:52	0:04:52	0:04:52	0:04:52

Y_train_dim (4000, 8422) **Label Cardinality** 2.9568 **Training Samples/Label** 0.4749
Y_test_dim (1000, 8422) **Label Density** 3.5107E-04

ผลการทดลองข้อมูลชุดทดสอบ สำหรับข้อมูล 6000 ตัวอย่าง

ตาราง 4.34 : เปรียบเทียบผลข้อมูล Linear 1-6000 ชุดทดสอบ สำหรับสัดส่วนการลดมิติ 0.5 ระหว่างแบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับ แบบจำลองดั้งต้น

Random Method	Linear 1-6000	Testing Result											
XMLC-PAO Ratio	org	XMLC-PAO-1 : 0.5						XMLC-PAO-2 : 0.5					
Adam Learning	0.003	0.0015	0.003	0.006	0.008	0.01	0.02	0.0015	0.003	0.006	0.008	0.01	0.02
mrr	0.0479	0.0434	0.0296	0.0391	0.0446	0.0427	0.0376	0.0419	0.0390	0.0387	0.0420	0.0380	0.0382
recall@5	2.4258	1.9133	1.1844	1.9609	1.6005	1.6971	1.2049	1.4659	1.4214	1.7223	1.9791	1.5071	1.5708
recall@10	3.1996	2.5316	1.3558	2.4947	2.9498	2.0809	1.6304	2.6725	1.7449	2.1143	2.4347	2.3278	2.1688
recall@30	5.6247	5.4941	2.4360	4.1845	5.1086	3.7634	4.1543	5.3764	2.7195	3.1654	3.9975	3.7230	3.4304
recall@50	7.3459	7.6071	3.0312	5.5219	6.1358	4.7561	5.6441	6.8316	3.7097	3.9434	5.0596	4.4634	4.4243
recall@100	9.5139	10.8027	7.5330	9.8891	10.5613	8.0537	9.9841	10.3884	8.8126	8.3828	10.0039	8.0792	9.3261
ndcg@5	2.0498	1.8153	1.1074	1.7092	1.6414	1.7031	1.3453	1.5098	1.4104	1.4771	1.7462	1.3720	1.4730
ndcg@10	2.4492	2.1709	1.2179	1.9901	2.1795	1.8855	1.5900	1.9985	1.6033	1.7087	2.0049	1.8071	1.7577
ndcg@30	3.4661	3.2543	1.5833	2.6480	2.9422	2.4812	2.5009	3.0164	1.9906	2.2159	2.5270	2.3471	2.2511
ndcg@50	4.0583	3.8028	1.7710	3.1210	3.3494	2.8108	2.9296	3.4819	2.3094	2.5353	2.8757	2.6159	2.5549
ndcg@100	4.6465	4.6193	2.9192	4.1846	4.4434	3.6621	3.9329	4.4577	3.6003	3.6576	4.1509	3.5230	3.7791
Process_data_time													
seconds	322.2234	320.0941	306.1104	307.4387	309.2927	310.5418	309.5992	350.0211	338.4714	341.0316	342.0850	344.7982	346.1831
hrs	0:05:22	0:05:20	0:05:06	0:05:07	0:05:09	0:05:11	0:05:10	0:05:50	0:05:38	0:05:41	0:05:42	0:05:45	0:05:46

Y_train_dim	(4800, 7560)	Label Cardinality	4.2481	Training Samples/Label	0.6349
Y_test_dim	(1200, 7560)	Label Density	5.6192E-04		

ตาราง 4.35 : เปรียบเทียบผลข้อมูล Linear 1-6000 ชุดทดสอบ สำหรับสัดส่วนการลดมิติ 0.8 ระหว่างแบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับ แบบจำลองดั้งต้น

Random Method	Linear 1-6000	Testing Result											
XMLC-PAO Ratio	org	XMLC-PAO-1 : 0.8						XMLC-PAO-2 : 0.8					
Adam Learning	0.003	0.0015	0.003	0.006	0.008	0.01	0.02	0.0015	0.003	0.006	0.008	0.01	0.02
mrr	0.0479	0.0440	0.0378	0.0189	0.0360	0.0382	0.0451	0.0328	0.0397	0.0422	0.0360	0.0416	0.0400
recall@5	2.4258	1.7870	1.7089	0.4623	1.5269	1.5587	2.0534	0.8705	1.6811	1.5607	1.2008	2.2419	1.5530
recall@10	3.1996	2.3465	2.1088	0.6019	2.1439	2.2433	2.5457	1.2859	2.1042	2.1466	1.9080	2.6384	2.1149
recall@30	5.6247	4.3393	3.3434	0.9160	4.1365	3.2775	4.3713	2.1822	4.3533	4.8881	3.1285	4.6322	4.2015
recall@50	7.3459	5.7099	4.7733	1.4554	5.5346	4.3883	5.8955	2.7013	5.8014	6.7253	4.0508	5.8822	5.4764
recall@100	9.5139	11.0809	7.8875	6.5271	6.7792	8.5272	9.9197	7.6068	10.2043	9.8213	8.6905	10.1134	9.5416
ndcg@5	2.0498	1.7402	1.5265	0.4750	1.3880	1.4801	1.8297	0.9953	1.5841	1.6876	1.2277	1.8967	1.4769
ndcg@10	2.4492	2.0759	1.7511	0.5827	1.7967	1.7916	2.1212	1.2116	1.8167	2.0014	1.5392	2.1031	1.7964
ndcg@30	3.4661	2.9323	2.3176	0.7697	2.6373	2.2828	2.7703	1.5506	2.6485	2.9510	2.0371	2.8352	2.6545
ndcg@50	4.0583	3.3542	2.8259	0.9469	3.0998	2.6724	3.2263	1.7298	3.1838	3.4604	2.3512	3.2005	3.1179
ndcg@100	4.6465	4.5723	3.6387	2.1664	3.4929	3.7528	4.2941	2.9416	4.2994	4.2423	3.4891	4.3096	4.1756
Process_data_time													
seconds	322.2234	313.9615	314.3303	313.3186	314.4631	317.1575	314.7034	350.1158	349.9839	357.2913	353.7105	354.6048	354.0771
hrs	0:05:22	0:05:14	0:05:14	0:05:13	0:05:14	0:05:17	0:05:15	0:05:50	0:05:50	0:05:57	0:05:54	0:05:55	0:05:54

Y_train_dim	(4800, 7560)	Label Cardinality	4.2481	Training Samples/Label	0.6349
Y_test_dim	(1200, 7560)	Label Density	5.6192E-04		

ตาราง 4.36 : เปรียบเทียบผลข้อมูล Linear 1-6000 ชุดทดสอบ สำหรับสัดส่วนการลดมิติ 1.0 ระหว่างแบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับ แบบจำลองตั้งต้น

Random Method	Linear 1-6000	Testing Result											
XMLC-PAO Ratio	org	XMLC-PAO-1 : 1.0						XMLC-PAO-2 : 1.0					
Adam Learning	0.003	0.0015	0.003	0.006	0.008	0.01	0.02	0.0015	0.003	0.006	0.008	0.01	0.02
mrr	0.0479	0.0464	0.0368	0.0416	0.0396	0.0318	0.0455	0.0429	0.0415	0.0355	0.0321	0.0244	0.0403
recall@5	2.4258	2.4053	1.7131	1.8424	1.2715	1.1217	2.1311	1.5689	1.3965	1.4376	1.1308	0.6581	1.7985
recall@10	3.1996	3.6115	2.4451	2.4175	1.8425	1.6144	3.2484	2.4385	1.7658	1.7982	1.4896	0.8972	2.6508
recall@30	5.6247	5.6575	4.0421	4.0772	3.7245	2.3946	5.3148	4.2046	3.6731	3.1536	2.0910	1.7801	4.0281
recall@50	7.3459	7.0913	5.1999	6.0532	5.2096	2.9001	7.1944	5.3967	4.7728	4.4398	2.7005	2.3210	5.2962
recall@100	9.5139	9.9433	8.7022	10.0373	9.9584	7.5045	9.5367	9.2959	9.0769	9.3159	7.6073	7.7256	7.5245
ndcg@5	2.0498	2.0293	1.4698	1.7222	1.3942	1.1131	1.8112	1.6094	1.5095	1.3180	1.1442	0.6618	1.6096
ndcg@10	2.4492	2.5429	1.8217	2.0317	1.6853	1.3472	2.3718	1.9772	1.7111	1.5030	1.3341	0.8207	1.9941
ndcg@30	3.4661	3.3031	2.3445	2.7130	2.3514	1.6472	3.2785	2.5794	2.4252	2.0444	1.6284	1.1642	2.5239
ndcg@50	4.0583	3.7364	2.6776	3.3127	2.7857	1.8549	3.8233	3.0266	2.8278	2.4363	1.8393	1.3640	2.9717
ndcg@100	4.6465	4.5020	3.5928	4.3078	3.9446	2.9827	4.5292	4.0647	3.9481	3.6107	3.0380	2.6549	3.6681
Process_data_time													
seconds	322.2234	319.6888	320.3034	320.6595	322.1877	322.5688	322.2798	358.3766	359.9477	359.9815	362.3375	365.7862	364.7638
hrs	0:05:22	0:05:20	0:05:20	0:05:21	0:05:22	0:05:23	0:05:22	0:05:58	0:06:00	0:06:00	0:06:02	0:06:06	0:06:05

Y_train_dim	(4800, 7560)	Label Cardinality	4.2481	Training Samples/Label	0.6349
Y_test_dim	(1200, 7560)	Label Density	5.6192E-04		

ตาราง 4.37 : เปรียบเทียบผลข้อมูล Linear 6001-12000 ชุดทดสอบ สำหรับสัดส่วนการลดมิติ 0.5 ระหว่างแบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับ แบบจำลองดั้งต้น

Random Method	Linear 6001-12000	Testing Result											
XMLC-PAO Ratio	org	XMLC-PAO-1 : 0.5						XMLC-PAO-2 : 0.5					
Adam Learning	0.003	0.0015	0.003	0.006	0.008	0.01	0.02	0.0015	0.003	0.006	0.008	0.01	0.02
mrr	0.0368	0.0315	0.0308	0.0333	0.0336	0.0325	0.0348	0.0326	0.0323	0.0344	0.0337	0.0339	0.0320
recall@5	1.7802	1.4533	1.5222	1.8040	1.8711	1.7750	1.5667	1.5021	1.9594	1.7044	1.5742	1.6937	1.5887
recall@10	2.6970	2.2355	2.5785	2.5128	2.8238	2.6655	2.6931	2.2887	2.7936	2.3821	2.4968	2.5023	2.4496
recall@30	6.5862	4.8815	4.7556	5.0548	5.2081	4.4539	5.5661	4.3791	5.1335	5.1352	4.9098	4.9529	4.7108
recall@50	8.9812	6.5863	6.2093	7.0763	6.7891	5.9880	6.6747	6.0740	6.3390	6.8936	6.6176	6.7554	6.6048
recall@100	13.0578	9.4571	9.2530	9.4994	10.3764	9.3421	9.0932	9.3217	9.9706	10.1734	10.2647	9.2304	9.7849
ndcg@5	1.5841	1.3263	1.3228	1.5466	1.6052	1.5649	1.4736	1.4369	1.6135	1.5481	1.4171	1.5556	1.4335
ndcg@10	1.9659	1.6472	1.7380	1.8762	2.0000	1.9400	1.9531	1.7541	1.9451	1.8336	1.7861	1.8822	1.7872
ndcg@30	3.0876	2.4542	2.4038	2.6904	2.7124	2.4904	2.8269	2.3979	2.6339	2.6745	2.4932	2.6060	2.5196
ndcg@50	3.6831	2.8826	2.7896	3.1558	3.1313	2.8748	3.0943	2.8135	2.9429	3.1040	2.8890	3.0639	2.9944
ndcg@100	4.5987	3.5808	3.4582	3.7095	3.9082	3.6363	3.6355	3.5393	3.7266	3.8134	3.6883	3.5834	3.6673
Process_data_time													
seconds	330.6347	314.0862	313.1634	314.4717	315.2333	316.7370	316.9835	341.7964	328.9384	330.5347	330.3362	332.1372	332.7932
hrs	0:05:31	0:05:14	0:05:13	0:05:14	0:05:15	0:05:17	0:05:17	0:05:42	0:05:29	0:05:31	0:05:30	0:05:32	0:05:33

Y_train_dim	(4800, 7914)	Label Cardinality	3.0110	Training Samples/Label	0.6065
Y_test_dim	(1200, 7914)	Label Density	3.8047E-04		

ตาราง 4.38 : เปรียบเทียบผลข้อมูล Linear 6001-12000 ชุดทดสอบ สำหรับสัดส่วนการลดมิติ 0.8 ระหว่างแบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับ แบบจำลองดั้งต้น

Random Method	Linear 6001-12000	Testing Result											
XMLC-PAO Ratio	org	XMLC-PAO-1 : 0.8						XMLC-PAO-2 : 0.8					
Adam Learning	0.003	0.0015	0.003	0.006	0.008	0.01	0.02	0.0015	0.003	0.006	0.008	0.01	0.02
mrr	0.0368	0.0316	0.0338	0.0337	0.0290	0.0353	0.0272	0.0354	0.0324	0.0368	0.0265	0.0316	0.0329
recall@5	1.7802	1.3751	1.7418	1.7326	1.4820	1.7689	1.2199	1.9819	1.6460	1.5851	1.4101	1.6293	1.4346
recall@10	2.6970	2.5503	2.5989	3.1312	2.8364	2.5989	2.2864	3.1544	2.3633	2.6678	2.5483	2.8372	2.0018
recall@30	6.5862	5.1379	5.3673	5.6508	4.8975	4.8896	4.2926	5.9245	5.0126	4.8934	4.9695	4.9922	4.0035
recall@50	8.9812	7.0782	7.6546	7.3974	7.0887	6.8406	6.6108	7.9355	6.6659	7.7501	7.1215	6.6701	6.0684
recall@100	13.0578	10.5334	10.8419	10.1496	10.3500	9.3492	9.1441	11.0301	10.3025	11.1514	10.5513	9.9794	10.2457
ndcg@5	1.5841	1.2876	1.4789	1.4884	1.1432	1.6417	1.1114	1.7155	1.5081	1.5449	1.0729	1.4231	1.4402
ndcg@10	1.9659	1.7562	1.8585	2.0700	1.7038	1.9909	1.5421	2.2030	1.8061	1.9986	1.5111	1.9290	1.6734
ndcg@30	3.0876	2.5290	2.7413	2.8449	2.3479	2.6697	2.1843	3.0345	2.5671	2.6908	2.2652	2.5970	2.3095
ndcg@50	3.6831	2.9837	3.2950	3.2879	2.8936	3.1450	2.7225	3.4989	2.9828	3.3815	2.8323	2.9986	2.8090
ndcg@100	4.5987	3.7007	3.9793	3.8800	3.6283	3.6804	3.2940	4.1341	3.7828	4.1431	3.5759	3.7105	3.6899
Process_data_time													
seconds	330.6347	320.4890	320.2704	322.1011	322.4132	324.0909	324.4111	342.2251	342.9915	344.7878	344.9005	346.0650	346.9204
hrs	0:05:31	0:05:20	0:05:20	0:05:22	0:05:22	0:05:24	0:05:24	0:05:42	0:05:43	0:05:45	0:05:45	0:05:46	0:05:47

Y_train_dim	(4800, 7914)	Label Cardinality	3.0110	Training Samples/Label	0.6065
Y_test_dim	(1200, 7914)	Label Density	3.8047E-04		

ตาราง 4.39 : เปรียบเทียบผลข้อมูล Linear 6001-12000 ชุดทดสอบ สำหรับสัดส่วนการลดมิติ 1.0 ระหว่างแบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับ แบบจำลองตั้งต้น

Random Method	Linear 6001-12000	Testing Result											
XMLC-PAO Ratio	org	XMLC-PAO-1 : 1.0						XMLC-PAO-2 : 1.0					
Adam Learning	0.003	0.0015	0.003	0.006	0.008	0.01	0.02	0.0015	0.003	0.006	0.008	0.01	0.02
mrr	0.0368	0.0317	0.0337	0.0349	0.0274	0.0335	0.0332	0.0332	0.0325	0.0325	0.0345	0.0320	0.0311
recall@5	1.7802	1.3676	1.9181	2.0400	1.7052	1.8009	1.5537	1.7481	1.7634	1.6364	1.5486	1.6108	1.4718
recall@10	2.6970	2.6137	2.9925	3.0919	2.6297	2.7015	2.5587	2.8059	2.5961	2.2672	2.4634	2.4787	2.2419
recall@30	6.5862	5.7290	5.8589	6.1468	4.7895	5.5076	4.9746	5.3423	5.8258	4.9794	5.1678	4.4406	4.3543
recall@50	8.9812	7.3066	7.5731	7.6646	7.3674	6.7564	7.0952	7.2080	7.6982	7.3242	7.1415	6.5655	6.3645
recall@100	13.0578	10.1668	10.0098	10.9631	9.8303	9.6582	10.4437	10.3685	11.0949	10.8849	9.7646	9.8487	8.8648
ndcg@5	1.5841	1.3004	1.5749	1.6895	1.3076	1.5523	1.3718	1.4966	1.4374	1.5312	1.4750	1.4640	1.3605
ndcg@10	1.9659	1.8039	2.0372	2.0882	1.6955	1.9279	1.7924	1.9190	1.8015	1.8020	1.8788	1.8265	1.6706
ndcg@30	3.0876	2.7021	2.9170	2.9979	2.3565	2.7810	2.5628	2.7266	2.7800	2.6667	2.6675	2.4138	2.3311
ndcg@50	3.6831	3.0767	3.3458	3.3829	3.0015	3.1171	3.0671	3.1909	3.2420	3.2279	3.1494	2.9219	2.8300
ndcg@100	4.5987	3.6895	3.9145	4.0505	3.5596	3.7431	3.7854	3.8784	3.9632	3.9727	3.7298	3.6489	3.3678
Process_data_time													
seconds	330.6347	327.6195	328.0285	327.9177	328.6652	330.4154	330.7495	351.4285	352.9922	352.4577	353.9505	356.0543	355.4368
hrs	0:05:31	0:05:28	0:05:28	0:05:28	0:05:29	0:05:30	0:05:31	0:05:51	0:05:53	0:05:52	0:05:54	0:05:56	0:05:55

Y_train_dim	(4800, 7914)	Label Cardinality	3.0110	Training Samples/Label	0.6065
Y_test_dim	(1200, 7914)	Label Density	3.8047E-04		

ตาราง 4.40 : เปรียบเทียบผลข้อมูล Linear 12001-18000 ชุดทดสอบ สำหรับสัดส่วนการลดมิติ 0.5 ระหว่างแบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับ แบบจำลองตั้งต้น

Random Method	Linear 12001-18000	Testing Result											
XMLC-PAO Ratio	org	XMLC-PAO-1 : 0.5						XMLC-PAO-2 : 0.5					
Adam Learning	0.003	0.0015	0.003	0.006	0.008	0.01	0.02	0.0015	0.003	0.006	0.008	0.01	0.02
mrr	0.0318	0.0276	0.0307	0.0272	0.0297	0.0218	0.0250	0.0241	0.0303	0.0259	0.0252	0.0244	0.0162
recall@5	2.0849	1.0787	1.6676	1.6855	2.0170	1.4158	1.4062	1.2840	1.9383	1.9083	1.6169	1.7585	1.2404
recall@10	3.4380	1.8974	2.4359	2.4322	2.8099	2.2388	2.2125	2.0608	2.8831	2.7414	2.3627	2.3839	1.7355
recall@30	6.1117	5.0816	4.3980	4.4237	4.6400	4.5281	5.0343	5.1006	4.8496	4.6450	4.2251	4.2312	3.9293
recall@50	7.4057	5.9806	5.7478	5.4698	5.9611	5.4961	6.4286	6.1877	5.8216	5.9273	5.6732	5.1448	5.0903
recall@100	10.7713	7.9297	7.5406	8.2847	8.0417	6.9133	8.7792	8.4303	8.0322	7.7832	8.3641	7.4190	6.9788
ndcg@5	1.5512	1.0749	1.5264	1.3450	1.5303	1.0739	1.1611	1.0371	1.6172	1.3908	1.3202	1.2969	0.8292
ndcg@10	2.0728	1.3966	1.8344	1.6394	1.8483	1.3998	1.4911	1.3467	2.0039	1.7234	1.6150	1.5768	1.0412
ndcg@30	2.8464	2.2974	2.4357	2.2218	2.3665	2.0481	2.3053	2.2170	2.5971	2.3027	2.1478	2.1105	1.6960
ndcg@50	3.1838	2.5270	2.7781	2.4881	2.6900	2.2811	2.6574	2.4793	2.8406	2.6085	2.4940	2.3310	1.9975
ndcg@100	3.8877	2.9194	3.1546	3.0848	3.1430	2.5977	3.1464	2.9931	3.3186	3.0250	3.0453	2.8088	2.3916
Process_data_time													
seconds	340.4285	334.3774	321.5686	322.6687	325.6223	325.6117	327.0249	333.3282	322.7012	322.9989	324.4087	325.3682	326.3478
hrs	0:05:40	0:05:34	0:05:22	0:05:23	0:05:26	0:05:26	0:05:27	0:05:33	0:05:23	0:05:23	0:05:24	0:05:25	0:05:26

Y_train_dim	(4800, 8109)	Label Cardinality	2.6469	Training Samples/Label	0.5919
Y_test_dim	(1200, 8109)	Label Density	3.2641E-04		

ตาราง 4.41 : เปรียบเทียบผลข้อมูล Linear 12001-18000 ชุดทดสอบ สำหรับสัดส่วนการลดมิติ 0.8 ระหว่างแบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับ แบบจำลองตั้งต้น

Random Method	Linear 12001-18000	Testing Result											
XMLC-PAO Ratio	org	XMLC-PAO-1 : 0.8						XMLC-PAO-2 : 0.8					
Adam Learning	0.003	0.0015	0.003	0.006	0.008	0.01	0.02	0.0015	0.003	0.006	0.008	0.01	0.02
mrr	0.0318	0.0241	0.0296	0.0281	0.0204	0.0276	0.0280	0.0281	0.0242	0.0285	0.0292	0.0239	0.0243
recall@5	2.0849	1.2148	1.8367	1.5294	1.4661	2.0765	1.4506	1.3289	1.5234	1.6470	1.7426	1.4564	1.3855
recall@10	3.4380	1.9216	2.6532	2.2896	2.3817	2.9088	2.3535	2.4497	2.0001	2.6258	2.3350	2.3783	1.9310
recall@30	6.1117	3.6657	4.5729	4.3592	3.9166	4.1727	4.7274	3.8473	3.5422	4.4070	5.0646	4.3541	3.7139
recall@50	7.4057	5.0380	5.5856	6.0409	5.4587	4.7977	5.8957	5.1378	4.5102	5.1538	6.1529	5.1445	5.3551
recall@100	10.7713	7.3167	7.7693	8.0887	8.7014	7.3036	7.5629	7.8002	6.3960	6.4588	8.5158	7.6242	7.3952
ndcg@5	1.5512	1.0239	1.5561	1.2967	1.1752	1.5303	1.2425	1.2046	1.3224	1.3496	1.3965	1.1514	1.1961
ndcg@10	2.0728	1.3282	1.8860	1.5778	1.5576	1.8780	1.5924	1.6234	1.5054	1.7329	1.6549	1.4851	1.3998
ndcg@30	2.8464	1.8412	2.4372	2.1858	1.9951	2.2614	2.2801	2.0164	1.9982	2.2703	2.4764	2.0572	1.9570
ndcg@50	3.1838	2.1794	2.6789	2.5905	2.3526	2.4157	2.5713	2.3172	2.2485	2.4531	2.7239	2.2574	2.3335
ndcg@100	3.8877	2.6709	3.1461	3.0426	3.0221	2.9467	2.9259	2.8455	2.6450	2.7416	3.2062	2.7998	2.7587
Process_data_time													
seconds	340.4285	330.3542	330.9353	333.1282	332.1346	333.6525	333.7004	336.2830	337.6706	338.7178	338.6511	340.4044	340.6453
hrs	0:05:40	0:05:30	0:05:31	0:05:33	0:05:32	0:05:34	0:05:34	0:05:36	0:05:38	0:05:39	0:05:39	0:05:40	0:05:41

Y_train_dim	(4800, 8109)	Label Cardinality	2.6469	Training Samples/Label	0.5919
Y_test_dim	(1200, 8109)	Label Density	3.2641E-04		

ตาราง 4.42 : เปรียบเทียบผลข้อมูล Linear 12001-18000 ชุดทดสอบ สำหรับสัดส่วนการลดมิติ 1.0 ระหว่างแบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับ แบบจำลองตั้งต้น

Random Method	Linear 12001-18000	Testing Result											
XMLC-PAO Ratio	org	XMLC-PAO-1 : 1.0						XMLC-PAO-2 : 1.0					
Adam Learning	0.003	0.0015	0.003	0.006	0.008	0.01	0.02	0.0015	0.003	0.006	0.008	0.01	0.02
mrr	0.0318	0.0289	0.0271	0.0298	0.0237	0.0217	0.0273	0.0288	0.0252	0.0261	0.0246	0.0279	0.0235
recall@5	2.0849	2.0689	1.7339	1.7529	1.4744	1.3213	1.9335	2.0251	1.1196	1.4288	1.7460	1.6112	1.1447
recall@10	3.4380	2.9200	2.9840	2.6287	2.7508	1.8557	2.5421	2.5940	1.7496	2.1242	2.6476	2.2779	1.6692
recall@30	6.1117	4.5107	5.1460	4.5600	4.3496	3.5806	4.4342	3.9592	3.7971	4.3098	4.9654	4.3544	4.4476
recall@50	7.4057	5.2060	6.1591	5.7643	5.4817	5.0519	5.5169	5.1523	5.3125	5.8342	6.1348	6.5798	5.4554
recall@100	10.7713	7.5142	8.1707	8.6540	7.2148	7.6388	7.8786	7.0128	7.4255	8.9133	8.1387	8.9567	7.1687
ndcg@5	1.5512	1.5229	1.2814	1.5310	1.1713	1.0670	1.5214	1.5965	1.0469	1.2014	1.2659	1.3473	0.9792
ndcg@10	2.0728	1.8454	1.7491	1.9072	1.6285	1.2948	1.7809	1.8433	1.3028	1.4677	1.6059	1.6053	1.2117
ndcg@30	2.8464	2.3174	2.4140	2.5047	2.0892	1.8114	2.3507	2.2432	1.8715	2.1270	2.2968	2.2470	2.0019
ndcg@50	3.1838	2.4859	2.6761	2.7787	2.3769	2.1378	2.6267	2.5332	2.2151	2.4838	2.5837	2.7772	2.2739
ndcg@100	3.8877	2.9917	3.1186	3.3819	2.7399	2.6753	3.1078	2.9295	2.6744	3.1115	3.0000	3.2649	2.6630
Process_data_time													
seconds	340.4285	335.1420	338.6070	337.9653	339.4068	341.1133	340.4696	344.3207	345.6679	345.6973	347.3022	349.6038	348.6850
hrs	0:05:40	0:05:35	0:05:39	0:05:38	0:05:39	0:05:41	0:05:40	0:05:44	0:05:46	0:05:46	0:05:47	0:05:50	0:05:49

Y_train_dim	(4800, 8109)	Label Cardinality	2.6469	Training Samples/Label	0.5919
Y_test_dim	(1200, 8109)	Label Density	3.2641E-04		

ตาราง 4.43 : เปรียบเทียบผลข้อมูล Linear 18001-24000 ชุดทดสอบ สำหรับสัดส่วนการลดมิติ 0.5 ระหว่างแบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับ แบบจำลองตั้งต้น

Random Method	Linear 18001-24000	Testing Result											
XMLC-PAO Ratio	org	XMLC-PAO-1 : 0.5						XMLC-PAO-2 : 0.5					
Adam Learning	0.003	0.0015	0.003	0.006	0.008	0.01	0.02	0.0015	0.003	0.006	0.008	0.01	0.02
mrr	0.0250	0.0192	0.0246	0.0150	0.0202	0.0173	0.0157	0.0172	0.0193	0.0137	0.0188	0.0176	0.0181
recall@5	1.8577	1.0922	1.7729	0.6415	1.5587	0.9957	0.6207	0.7901	1.0859	1.2958	1.0705	1.0390	0.7832
recall@10	2.4809	1.7596	2.5938	1.1021	2.2698	2.0578	1.2644	1.7118	1.8693	2.2945	1.9974	2.0946	1.7651
recall@30	5.1728	3.6030	4.5580	3.1384	4.0503	4.1666	3.1590	3.7852	4.3815	3.8466	4.0837	4.3986	4.4596
recall@50	7.3459	5.1763	5.6061	4.2933	4.7819	5.1376	5.0673	4.6061	5.9392	4.6615	5.3333	5.5092	5.9786
recall@100	10.7023	7.5862	7.6082	6.0818	6.7405	6.7980	6.9194	5.9186	7.5884	6.5991	7.4902	7.8023	8.3181
ndcg@5	1.3152	0.9144	1.3225	0.5593	1.0858	0.8074	0.5529	0.6753	0.8413	0.7270	0.8374	0.7765	0.6647
ndcg@10	1.5730	1.1781	1.6336	0.7410	1.3464	1.1911	0.8270	1.0447	1.1194	1.1260	1.1894	1.1533	1.0422
ndcg@30	2.3363	1.6991	2.2154	1.3532	1.8578	1.8247	1.3679	1.6306	1.8643	1.5856	1.7795	1.8430	1.8185
ndcg@50	2.8266	2.0594	2.4812	1.6061	2.0411	2.0585	1.7906	1.8351	2.2060	1.7772	2.0555	2.0958	2.1596
ndcg@100	3.5441	2.5354	2.9018	1.9776	2.4586	2.4042	2.1723	2.1011	2.5732	2.1504	2.4861	2.5444	2.6162
Process_data_time													
seconds	330.1702	311.5902	313.0920	313.6789	314.8519	315.4998	316.0043	326.9791	329.2990	329.9521	332.0146	331.7797	332.5393
hrs	0:05:30	0:05:12	0:05:13	0:05:14	0:05:15	0:05:15	0:05:16	0:05:27	0:05:29	0:05:30	0:05:32	0:05:32	0:05:33

Y_train_dim	(4800, 8116)	Label Cardinality	2.4935	Training Samples/Label	0.5914
Y_test_dim	(1200, 8116)	Label Density	3.0724E-04		

ตาราง 4.44 : เปรียบเทียบผลข้อมูล Linear 18001-24000 ชุดทดสอบ สำหรับสัดส่วนการลดมิติ 0.8 ระหว่างแบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับ แบบจำลองตั้งต้น

Random Method	Linear 18001-24000	Testing Result											
XMLC-PAO Ratio	org	XMLC-PAO-1 : 0.8						XMLC-PAO-2 : 0.8					
Adam Learning	0.003	0.0015	0.003	0.006	0.008	0.01	0.02	0.0015	0.003	0.006	0.008	0.01	0.02
mrr	0.0250	0.0187	0.0124	0.0205	0.0161	0.0185	0.0231	0.0128	0.0233	0.0201	0.0257	0.0280	0.0217
recall@5	1.8577	1.2022	0.9123	1.0946	1.0872	1.2454	1.7687	0.9795	1.8354	1.0537	1.6396	2.1604	1.5799
recall@10	2.4809	2.2224	1.6208	2.1606	2.1304	2.0267	2.5302	1.8041	2.1049	1.8593	2.7202	3.0794	2.1347
recall@30	5.1728	3.9414	3.4532	4.7330	4.1628	4.3076	3.9242	3.3876	3.9877	3.6131	4.8339	4.7078	4.3829
recall@50	7.3459	5.1935	4.4477	6.1930	5.2304	5.0021	4.9069	4.9043	5.2853	5.2469	5.8124	5.8641	6.0564
recall@100	10.7023	6.6611	6.0628	8.2343	7.1715	6.6749	6.5730	6.4814	7.5153	7.4663	7.0632	7.5905	7.7578
ndcg@5	1.3152	0.8880	0.6698	0.8633	0.7877	0.9177	1.2576	0.6157	1.3510	0.8528	1.2930	1.5912	1.1536
ndcg@10	1.5730	1.2690	0.9316	1.2510	1.1511	1.2329	1.5756	0.9302	1.4679	1.1719	1.7258	1.9501	1.3676
ndcg@30	2.3363	1.7943	1.4849	2.0088	1.7128	1.8795	2.0120	1.4028	2.0375	1.6959	2.3089	2.4077	1.9923
ndcg@50	2.8266	2.0711	1.7091	2.3461	1.9557	2.0404	2.2532	1.7566	2.3597	2.0643	2.5369	2.6718	2.3666
ndcg@100	3.5441	2.3657	2.0471	2.7488	2.3353	2.3744	2.5882	2.0868	2.8116	2.5057	2.8220	3.0490	2.7389
Process_data_time													
seconds	330.1702	319.1146	320.3805	321.6605	321.8609	323.3271	324.2843	343.1406	343.6759	345.9554	345.9595	347.5853	347.2350
hrs	0:05:30	0:05:19	0:05:20	0:05:22	0:05:22	0:05:23	0:05:24	0:05:43	0:05:44	0:05:46	0:05:46	0:05:48	0:05:47

Y_train_dim	(4800, 8116)	Label Cardinality	2.4935	Training Samples/Label	0.5914
Y_test_dim	(1200, 8116)	Label Density	3.0724E-04		

ตาราง 4.45 : เปรียบเทียบผลข้อมูล Linear 18001-24000 ชุดทดสอบ สำหรับสัดส่วนการลดมิติ 1.0 ระหว่างแบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับ แบบจำลองตั้งต้น

Random Method	Linear 18001-24000	Testing Result											
XMLC-PAO Ratio	org	XMLC-PAO-1 : 1.0						XMLC-PAO-2 : 1.0					
Adam Learning	0.003	0.0015	0.003	0.006	0.008	0.01	0.02	0.0015	0.003	0.006	0.008	0.01	0.02
mrr	0.0250	0.0126	0.0237	0.0152	0.0231	0.0210	0.0160	0.0175	0.0190	0.0176	0.0235	0.0213	0.0211
recall@5	1.8577	0.9550	1.8778	0.9927	1.6443	1.5984	0.7717	1.2188	1.1415	0.8679	1.6388	1.1950	1.4540
recall@10	2.4809	1.5989	3.0709	1.7326	2.2187	2.3813	2.2725	2.1695	2.1632	1.7481	2.7374	2.2329	2.2715
recall@30	5.1728	3.9290	4.9718	4.0249	4.9482	3.7677	3.6354	3.8743	4.2666	3.5779	4.6423	4.2690	4.1250
recall@50	7.3459	5.5409	6.1190	4.7874	6.2006	5.0177	4.9048	5.3503	5.6960	5.6218	5.7666	5.8144	5.5040
recall@100	10.7023	6.9599	7.1449	7.0202	8.2932	6.7887	6.6266	6.6978	8.1260	8.5088	7.0538	7.7743	7.5908
ndcg@5	1.3152	0.6154	1.3489	0.7827	1.2368	1.1419	0.6321	0.8668	0.8250	0.7598	1.2010	0.9918	1.0774
ndcg@10	1.5730	0.8728	1.7981	1.0827	1.4808	1.4462	1.2139	1.2124	1.2295	1.0903	1.6036	1.3703	1.3785
ndcg@30	2.3363	1.5479	2.3569	1.7352	2.2193	1.8353	1.6230	1.7112	1.8673	1.5914	2.1747	1.9751	1.9400
ndcg@50	2.8266	1.9204	2.6139	1.9218	2.4902	2.1054	1.9306	2.0521	2.1704	2.0661	2.4481	2.3469	2.2640
ndcg@100	3.5441	2.2161	2.8264	2.3710	2.9201	2.4602	2.3021	2.3368	2.6651	2.6118	2.7246	2.7530	2.6907
Process_data_time													
seconds	330.1702	326.1686	327.9499	328.5812	328.7279	330.8992	330.8118	351.4135	354.5476	353.6833	354.2561	357.0741	355.8982
hrs	0:05:30	0:05:26	0:05:28	0:05:29	0:05:29	0:05:31	0:05:31	0:05:51	0:05:55	0:05:54	0:05:54	0:05:57	0:05:56

Y_train_dim	(4800, 8116)	Label Cardinality	2.4935	Training Samples/Label	0.5914
Y_test_dim	(1200, 8116)	Label Density	3.0724E-04		

ตาราง 4.46 : เปรียบเทียบผลข้อมูล Linear 24001-27932 (3933 ตัวอย่าง) ชุดทดสอบ สำหรับสัดส่วนการลดมิติ 0.5 ระหว่างแบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับแบบจำลองตั้งต้น

Random Method	Linear 24001-27932	Testing Result											
XMLC-PAO Ratio	org	XMLC-PAO-1 : 0.5						XMLC-PAO-2 : 0.5					
Adam Learning	0.003	0.0015	0.003	0.006	0.008	0.01	0.02	0.0015	0.003	0.006	0.008	0.01	0.02
mrr	0.0153	0.0146	**0.0169**	0.0139	0.0139	0.0153	0.0119	0.0103	0.0121	0.0150	0.0136	0.0152	0.0141
recall@5	0.7730	0.8365	1.0165	0.7276	0.7276	**1.1012**	0.5930	0.5400	0.5612	0.6247	0.6396	0.7730	0.7488
recall@10	1.8573	1.1648	**2.0013**	1.5323	1.5323	1.7302	1.0208	0.8196	0.9742	0.9890	1.1902	1.2295	1.5260
recall@30	3.7508	1.9377	3.5372	3.3367	3.3367	**4.2560**	3.2873	2.0279	2.9989	2.3655	3.1927	3.3333	3.4129
recall@50	5.3648	3.5352	4.4130	4.3934	4.3934	**5.4601**	4.6183	3.3358	3.8640	3.6499	4.4209	4.7660	4.4442
recall@100	**9.0599**	6.5583	5.6752	6.7591	6.7591	7.0898	6.9469	5.1761	5.8134	6.0242	5.9158	7.3773	6.2402
ndcg@5	0.6165	0.7217	**0.8121**	0.6057	0.6057	0.7823	0.4902	0.4371	0.5293	0.6430	0.5529	0.6348	0.6344
ndcg@10	1.0350	0.8626	**1.1745**	0.8982	0.8982	1.0113	0.6796	0.5491	0.7006	0.7925	0.8005	0.8172	0.9280
ndcg@30	1.6107	1.1145	1.6518	1.3952	1.3952	**1.7154**	1.3159	0.8659	1.2631	1.1606	1.3547	1.4297	1.4390
ndcg@50	1.9969	1.4988	1.8640	1.6508	1.6508	**2.0038**	1.6259	1.1567	1.4781	1.4429	1.6162	1.7677	1.6689
ndcg@100	**2.6742**	2.0419	2.1314	2.1145	2.1145	2.3472	2.0843	1.5159	1.8825	1.9125	1.9308	2.2576	2.0506
Process_data_time													
seconds	225.7609	224.8722	**212.2504**	212.8361	212.5401	212.3979	213.6050	232.1737	219.4581	219.2147	220.9835	221.4392	222.1600
hrs	0:03:46	0:03:45	**0:03:32**	0:03:33	0:03:33	0:03:32	0:03:34	0:03:52	0:03:39	0:03:39	0:03:41	0:03:41	0:03:42

Y_train_dim	(3146, 5983)	Label Cardinality	2.4619	Training Samples/Label	0.5258
Y_test_dim	(787, 5983)	Label Density	4.1148E-04		

ตาราง 4.47 : เปรียบเทียบผลข้อมูล Linear 24001-27932 (3933 ตัวอย่าง) ชุดทดสอบ สำหรับสัดส่วนการลดมิติ 0.8 ระหว่างแบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับแบบจำลองตั้งต้น

Random Method	Linear 24001-27932	Testing Result											
XMLC-PAO Ratio	org	XMLC-PAO-1 : 0.8						XMLC-PAO-2 : 0.8					
Adam Learning	0.003	0.0015	0.003	0.006	0.008	0.01	0.02	0.0015	0.003	0.006	0.008	0.01	0.02
mrr	0.0153	0.0134	0.0155	0.0122	0.0145	0.0160	0.0139	**0.0160**	0.0157	0.0140	0.0127	0.0136	0.0137
recall@5	0.7730	0.5586	0.7730	0.7243	0.9953	0.8789	0.6565	**1.2918**	1.1330	0.6989	0.7412	1.2071	1.0271
recall@10	**1.8573**	0.9016	1.6984	1.2008	1.6603	1.6954	0.9890	1.3841	1.6171	1.0885	0.9318	1.6413	1.6624
recall@30	**3.7508**	2.0177	3.1302	3.1409	2.7862	3.3552	3.1835	2.7281	2.9915	1.9822	3.5315	3.1307	3.0508
recall@50	**5.3648**	3.2417	4.2399	4.1205	3.9795	3.7576	4.1419	3.2681	4.2949	3.1439	4.5994	4.3950	4.0800
recall@100	**9.0599**	5.9432	5.7795	4.9476	5.6791	5.6002	6.1020	4.8395	5.8345	4.6415	6.5302	6.8664	5.3040
ndcg@5	0.6165	0.5434	0.7054	0.5450	0.7637	0.7574	0.6620	**0.9802**	0.8715	0.6713	0.5898	0.8214	0.8271
ndcg@10	1.0350	0.6769	1.0399	0.7309	1.0248	1.0595	0.7998	1.0250	**1.0700**	0.8312	0.6675	1.0193	1.0537
ndcg@30	**1.6107**	0.9913	1.4303	1.3004	1.3658	1.5704	1.4293	1.3969	1.4348	1.0973	1.3446	1.4415	1.4783
ndcg@50	**1.9969**	1.2703	1.6781	1.5437	1.6568	1.6668	1.6440	1.5305	1.7414	1.3473	1.6000	1.7167	1.7396
ndcg@100	**2.6742**	1.8060	1.9935	1.7243	2.0021	2.0712	2.0550	1.8623	2.0479	1.6649	1.9918	2.1666	1.9885
Process_data_time													
seconds	225.7609	**216.0009**	216.5009	218.6226	218.7450	219.5671	218.1574	224.9017	225.1814	225.7146	226.7655	226.9348	228.7295
hrs	0:03:46	**0:03:36**	0:03:37	0:03:39	0:03:39	0:03:40	0:03:38	0:03:45	0:03:45	0:03:46	0:03:47	0:03:47	0:03:49

Y_train_dim	(3146, 5983)	Label Cardinality	2.4619	Training Samples/Label	0.5258
Y_test_dim	(787, 5983)	Label Density	4.1148E-04		

ตาราง 4.48 : เปรียบเทียบผลข้อมูล Linear 24001-27932 (3933 ตัวอย่าง) ชุดทดสอบ สำหรับสัดส่วนการลดมิติ 1.0 ระหว่างแบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับแบบจำลองตั้งต้น

Random Method	Linear 24001-27932	Testing Result											
XMLC-PAO Ratio	org	XMLC-PAO-1 : 1.0						XMLC-PAO-2 : 1.0					
Adam Learning	0.003	0.0015	0.003	0.006	0.008	0.01	0.02	0.0015	0.003	0.006	0.008	0.01	0.02
mrr	0.0153	0.0129	0.0134	0.0136	0.0113	0.0159	**0.0170**	0.0137	0.0152	0.0128	0.0152	0.0128	0.0135
recall@5	0.7730	0.5730	0.8683	0.9742	0.6141	0.7730	**1.2643**	0.7730	0.6989	1.0165	0.7412	0.8767	0.8789
recall@10	**1.8573**	0.7818	1.0313	1.6307	1.0589	1.2071	1.4655	1.3777	1.5396	1.1309	1.5883	1.0779	1.4549
recall@30	**3.7508**	2.5906	2.4495	3.6555	2.0330	2.9451	2.3925	2.6950	3.5360	2.2575	3.0220	2.1952	2.5942
recall@50	**5.3648**	3.5813	3.8197	5.1508	2.7637	4.3186	3.7468	4.2468	4.4467	3.1535	4.2939	3.4023	3.9203
recall@100	**9.0599**	6.5667	5.7175	6.9615	3.9339	5.7779	5.3564	6.2069	5.7704	4.6297	6.3917	6.1233	5.3656
ndcg@5	0.6165	0.5003	0.6872	0.6943	0.4992	0.7000	**0.9514**	0.6347	0.6474	0.7349	0.6548	0.6863	0.6585
ndcg@10	**1.0350**	0.6006	0.7650	0.9270	0.6856	0.8773	1.0331	0.8757	0.9370	0.7869	0.9700	0.7671	0.8827
ndcg@30	**1.6107**	1.0602	1.1460	1.4794	0.9703	1.3906	1.3162	1.2457	1.4890	1.1365	1.3612	1.0670	1.2417
ndcg@50	**1.9969**	1.3045	1.4594	1.8337	1.1274	1.7099	1.6405	1.5970	1.6763	1.3495	1.6426	1.3295	1.5465
ndcg@100	**2.6742**	1.9083	1.8358	2.1894	1.3781	2.0328	1.9673	1.9740	1.9525	1.6417	2.0563	1.8678	1.8249
Process_data_time													
seconds	225.7609	**219.6729**	221.3547	221.5515	222.9238	224.0557	225.0356	231.2841	233.6357	233.8867	233.5276	235.3800	236.5150
hrs	0:03:46	**0:03:40**	0:03:41	0:03:42	0:03:43	0:03:44	0:03:45	0:03:51	0:03:54	0:03:54	0:03:54	0:03:55	0:03:57

Y_train_dim	(3146, 5983)	Label Cardinality	2.4619	Training Samples/Label	0.5258
Y_test_dim	(787, 5983)	Label Density	4.1148E-04		

ผลการทดลองข้อมูลชุดทดสอบ สำหรับข้อมูล 7000 ตัวอย่าง

ตาราง 4.49 : เปรียบเทียบผลข้อมูล Linear 1-7000 ชุดทดสอบ สำหรับสัดส่วนการลดมิติ 0.5 ระหว่างแบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับ แบบจำลองดั้งต้น

Random Method	Linear 1-7000	Testing Result											
XMLC-PAO Ratio	org	XMLC-PAO-1 : 0.5						XMLC-PAO-2 : 0.5					
Adam Learning	0.003	0.0015	0.003	0.006	0.008	0.01	0.02	0.0015	0.003	0.006	0.008	0.01	0.02
mrr	0.0488	0.0415	0.0273	0.0443	0.0407	0.0380	0.0317	0.0461	0.0448	0.0400	0.0376	0.0385	0.0418
recall@5	2.4069	1.7582	1.1619	2.0557	1.8653	1.3236	1.1587	2.0160	1.9431	1.8505	1.4465	1.9875	2.0618
recall@10	3.1517	2.5320	1.7095	2.6059	2.3695	1.8420	1.5973	2.4934	2.6968	2.3463	1.6817	2.4350	2.7533
recall@30	4.8021	3.5823	2.4365	4.5268	4.5115	3.1101	2.6664	4.7775	4.2516	3.7794	2.6776	4.2345	5.0578
recall@50	7.0186	4.7678	3.0852	6.3064	5.4400	3.8390	3.7849	5.4525	6.4821	5.2935	3.7653	6.0195	7.0697
recall@100	10.5463	8.8068	7.2328	10.1642	8.9169	7.5830	7.1186	8.1938	10.0396	8.9733	7.8124	10.2810	10.1484
ndcg@5	2.1462	1.7314	1.0356	1.9196	1.7113	1.3575	1.2166	1.9915	1.8739	1.7427	1.5247	1.6602	1.8472
ndcg@10	2.5086	2.0622	1.3000	2.2362	1.9712	1.6068	1.4447	2.2695	2.2048	1.9974	1.6410	1.9305	2.1874
ndcg@30	3.2583	2.5277	1.6182	3.0004	2.7073	2.1054	1.8925	3.0491	2.8115	2.5674	2.0897	2.6617	2.9688
ndcg@50	3.9283	2.8498	1.8446	3.5429	3.0218	2.3805	2.2263	3.2897	3.4201	3.0089	2.4580	3.2090	3.5201
ndcg@100	4.7734	3.7993	2.7837	4.4496	3.8875	3.3210	3.0511	3.9445	4.2859	3.8909	3.4661	4.2171	4.2105
Process_data_time													
seconds	396.9514	384.4198	372.3381	374.0061	375.7066	376.0563	378.2971	391.6138	378.9385	380.4655	380.7068	386.5706	388.9956
hrs	0:06:37	0:06:24	0:06:12	0:06:14	0:06:16	0:06:16	0:06:18	0:06:32	0:06:19	0:06:20	0:06:21	0:06:27	0:06:29

Y_train_dim	(5600, 8465)	Label Cardinality	4.1248	Training Samples/Label	0.6615
Y_test_dim	(1400, 8465)	Label Density	4.8728E-04		

ตาราง 4.50 : เปรียบเทียบผลข้อมูล Linear 1-7000 ชุดทดสอบ สำหรับสัดส่วนการลดมิติ 0.8 ระหว่างแบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับ แบบจำลองดั้งต้น

Random Method	Linear 1-7000	Testing Result											
XMLC-PAO Ratio	org	XMLC-PAO-1 : 0.8						XMLC-PAO-2 : 0.8					
Adam Learning	0.003	0.0015	0.003	0.006	0.008	0.01	0.02	0.0015	0.003	0.006	0.008	0.01	0.02
mrr	0.0488	0.0285	0.0426	0.0429	0.0211	0.0428	0.0352	0.0332	0.0445	0.0408	0.0443	0.0417	0.0410
recall@5	2.4069	0.8712	1.3304	1.9670	0.5532	1.2926	1.4244	1.3424	2.0350	1.6087	1.9490	1.6328	1.7268
recall@10	3.1517	1.4518	2.1562	2.5442	0.7221	2.3676	1.9328	1.8538	2.6622	2.3146	2.5208	2.3399	2.0397
recall@30	4.8021	4.4445	4.5040	3.8014	1.3701	3.8732	2.6805	3.0373	4.2529	3.7694	4.1338	4.2852	3.8828
recall@50	7.0186	5.8467	5.7351	5.2611	1.7286	4.8923	3.3926	3.3661	5.7665	4.4118	6.0008	5.5893	4.6897
recall@100	10.5463	8.1708	7.9320	9.0783	6.1345	7.8206	6.9314	6.9538	8.6762	8.1552	9.5781	7.6376	8.8770
ndcg@5	2.1462	0.9226	1.5183	1.8341	0.5766	1.5503	1.4373	1.2536	1.9228	1.6687	1.9053	1.7273	1.6621
ndcg@10	2.5086	1.2505	1.9357	2.1396	0.6851	1.9969	1.6577	1.5035	2.2160	1.9649	2.1833	2.0293	1.8320
ndcg@30	3.2583	2.3306	2.7893	2.6184	0.9478	2.5427	2.0163	1.9302	2.9184	2.4775	2.7742	2.7404	2.5338
ndcg@50	3.9283	2.7679	3.1607	3.0697	1.0872	2.9141	2.2732	2.0571	3.3812	2.7307	3.3206	3.1680	2.8330
ndcg@100	4.7734	3.3853	3.7464	4.0267	2.1419	3.6811	3.0983	2.9749	4.1030	3.6155	4.2493	3.7452	3.8787
Process_data_time													
seconds	396.9514	380.5977	395.9614	383.9937	379.9317	382.0580	383.2983	400.6010	413.9514	404.6038	400.2305	403.3406	402.6027
hrs	0:06:37	0:06:21	0:06:36	0:06:24	0:06:20	0:06:22	0:06:23	0:06:41	0:06:54	0:06:45	0:06:40	0:06:43	0:06:43

Y_train_dim	(5600, 8465)	Label Cardinality	4.1248	Training Samples/Label	0.6615
Y_test_dim	(1400, 8465)	Label Density	4.8728E-04		

ตาราง 4.51 : เปรียบเทียบผลข้อมูล Linear 1-7000 ชุดทดสอบ สำหรับสัดส่วนการลดมิติ 1.0 ระหว่างแบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับ แบบจำลองตั้งต้น

Random Method	Linear 1-7000	Testing Result											
XMLC-PAO Ratio	org	XMLC-PAO-1 : 1.0						XMLC-PAO-2 : 1.0					
Adam Learning	0.003	0.0015	0.003	0.006	0.008	0.01	0.02	0.0015	0.003	0.006	0.008	0.01	0.02
mrr	0.0488	0.0390	0.0423	0.0448	0.0487	0.0355	0.0441	0.0479	0.0386	0.0243	0.0444	0.0465	0.0351
recall@5	2.4069	1.7364	1.9562	1.7914	2.3650	1.1649	1.7004	2.0315	1.4696	0.6276	1.8146	1.9796	1.2377
recall@10	3.1517	2.3040	2.4512	2.1701	3.2226	1.4821	2.0676	3.1478	1.9790	0.7950	2.1564	2.6467	1.7942
recall@30	4.8021	3.5089	4.0062	3.8314	5.4178	4.5838	4.3116	5.3498	4.0484	1.3413	3.7383	5.0798	2.7055
recall@50	7.0186	4.2108	5.2458	4.6489	7.0446	6.4235	5.4659	6.7585	5.4118	1.7468	4.2389	6.1847	3.4193
recall@100	10.5463	7.7581	8.5227	7.7687	9.6642	8.9504	8.9849	10.5743	9.3424	5.8314	9.1791	9.2947	7.3833
ndcg@5	2.1462	1.6286	1.7955	1.8554	2.1386	1.3226	1.7590	1.9235	1.5217	0.7325	1.8854	1.9156	1.2967
ndcg@10	2.5086	1.9184	2.0425	2.0920	2.5734	1.5281	1.9747	2.4590	1.7425	0.8556	2.1155	2.2399	1.5422
ndcg@30	3.2583	2.4579	2.6211	2.7658	3.3657	2.5897	2.7385	3.2895	2.4053	1.1665	2.7352	3.1568	1.9242
ndcg@50	3.9283	2.7222	3.0313	3.0047	3.8422	3.1168	3.1048	3.6824	2.7860	1.3382	2.9252	3.5299	2.1860
ndcg@100	4.7734	3.6001	3.8503	3.7826	4.5036	3.7773	3.9250	4.5541	3.8034	2.3695	4.0741	4.3392	3.1779
Process_data_time													
seconds	396.9514	386.5607	387.7175	391.6824	393.3755	395.8901	395.8560	403.0293	406.3500	408.3421	407.0961	409.3504	410.0739
hrs	0:06:37	0:06:21	0:06:36	0:06:24	0:06:20	0:06:22	0:06:23	0:06:41	0:06:54	0:06:45	0:06:40	0:06:43	0:06:43

Y_train_dim	(5600, 8465)	Label Cardinality	4.1248	Training Samples/Label	0.6615
Y_test_dim	(1400, 8465)	Label Density	4.8728E-04		

ผลการทดลองข้อมูลชุดทดสอบ สำหรับข้อมูลทั้งหมด

ตาราง 4.52 : เปรียบเทียบผลข้อมูล Linear 1-27932 ชุดทดสอบ สำหรับสัดส่วนการลดมิติ 0.5 ระหว่างแบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับ แบบจำลองดั้งต้น

Random Method	Linear 1-27932	Testing Result											
XMLC-PAO Ratio	org	XMLC-PAO-1 : 0.5						XMLC-PAO-2 : 0.5					
Adam Learning	0.003	0.0015	0.003	0.006	0.008	0.01	0.02	0.0015	0.003	0.006	0.008	0.01	0.02
mrr	0.0279	0.0237	0.0191	0.0218	0.0222	0.0215	0.0225	0.0254	0.0246	0.0216	0.0246	0.0253	0.0234
recall@5	1.2867	1.0891	0.9739	0.7636	0.9635	1.1836	0.8741	1.3663	1.2242	1.0461	1.1398	1.2237	1.0043
recall@10	2.1165	1.5132	1.4085	1.0625	1.3190	1.5528	1.4265	2.0346	1.6337	1.2851	1.6609	1.9516	1.2846
recall@30	4.8364	2.2637	2.6025	2.2467	3.0479	3.1903	2.6856	3.2449	2.7908	2.7366	2.9585	3.3790	2.4530
recall@50	6.4497	3.2223	3.4106	3.5543	3.6640	4.1915	3.4708	4.5276	3.6614	3.5590	3.8135	4.3637	3.6218
recall@100	9.4752	5.0747	5.4945	5.1410	5.2487	5.7459	4.8754	6.7777	5.5947	4.9117	5.4682	6.7039	5.4518
ndcg@5	1.1189	1.0395	0.8255	0.8641	0.9492	0.9497	0.9025	1.1435	1.1179	0.9448	1.0596	1.1308	0.9992
ndcg@10	1.4645	1.2168	1.0149	0.9964	1.0914	1.1186	1.1359	1.4331	1.2954	1.0557	1.2983	1.4141	1.1361
ndcg@30	2.2716	1.4569	1.3849	1.3695	1.5938	1.6118	1.5159	1.7978	1.6417	1.5042	1.7081	1.8086	1.5157
ndcg@50	2.6635	1.6903	1.5763	1.6856	1.7476	1.8516	1.7089	2.1079	1.8546	1.7074	1.9213	2.0446	1.7865
ndcg@100	3.3283	2.0713	2.0207	2.0320	2.0893	2.1793	2.0236	2.5843	2.2420	2.0060	2.2639	2.5264	2.1649
Process_data_time													
seconds	1589.9088	1527.5139	1542.6340	1540.6797	1541.4248	1556.6501	1548.8470	2065.3880	2059.6768	2055.1546	2076.4978	2071.2446	2620.3325
hrs	0:26:30	0:18:33	0:18:30	0:18:24	0:18:09	0:18:06	0:18:01	0:21:10	0:20:53	0:20:52	0:20:39	0:20:42	0:20:27

Y_train_dim (22345, 23335) Label Cardinality 3.0060 Training Samples/Label 0.9576
Y_test_dim (5587, 23335) Label Density 1.2882E-04

ตาราง 4.53 : เปรียบเทียบผลข้อมูล Linear 1-27932 ชุดทดสอบ สำหรับสัดส่วนการลดมิติ 0.8 ระหว่างแบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับ แบบจำลองดั้งต้น

Random Method	Linear 1-27932	Testing Result											
XMLC-PAO Ratio	org	XMLC-PAO-1 : 0.8						XMLC-PAO-2 : 0.8					
Adam Learning	0.003	0.0015	0.003	0.006	0.008	0.01	0.02	0.0015	0.003	0.006	0.008	0.01	0.02
mrr	0.0279	0.0258	0.0255	0.0254	0.0245	0.0250	0.0250	0.0253	0.0248	0.0215	0.0241	0.0251	0.0237
recall@5	1.2867	1.1705	1.1922	1.4060	0.9745	1.0267	1.2218	1.4098	1.2140	0.9677	1.0813	1.1858	1.0374
recall@10	2.1165	1.9127	1.7889	2.1543	1.5625	1.6901	1.8120	1.9126	1.6430	1.3140	1.6880	1.9289	1.3444
recall@30	4.8364	3.8191	2.9070	3.7920	3.0587	3.2399	2.8067	3.2479	2.7508	2.5331	2.9791	3.3687	2.5655
recall@50	6.4497	5.1615	3.8340	4.4619	4.2945	4.0819	4.0745	4.6251	3.6553	3.4820	3.7530	4.3066	3.3719
recall@100	9.4752	6.9886	5.5255	5.7725	6.0513	5.8761	6.2252	6.8305	5.5280	4.9027	5.4453	6.6081	5.1192
ndcg@5	1.1189	1.0822	1.1109	1.1641	0.9914	1.0192	1.1232	1.1651	1.1030	0.9117	1.0151	1.0998	1.0036
ndcg@10	1.4645	1.3697	1.3586	1.4664	1.2284	1.2637	1.3562	1.3870	1.2881	1.0729	1.2780	1.3843	1.1451
ndcg@30	2.2716	1.9347	1.7049	1.9946	1.6832	1.7687	1.6672	1.7908	1.6240	1.4593	1.6918	1.7897	1.5357
ndcg@50	2.6635	2.2416	1.9446	2.1753	1.9771	1.9792	1.9766	2.1173	1.8494	1.6929	1.8782	2.0155	1.7438
ndcg@100	3.3283	2.6431	2.3133	2.4563	2.3625	2.3718	2.4417	2.5883	2.2222	2.0146	2.2583	2.4962	2.0959
Process_data_time													
seconds	1589.9088	1608.5858	1611.3437	1585.3152	1607.1866	1609.5486	1610.5634	1976.7135	2021.4516	2001.7952	2004.8599	2011.1834	2488.9721
hrs	0:26:30	0:26:49	0:26:51	0:26:25	0:26:47	0:26:50	0:26:51	0:32:57	0:33:41	0:33:22	0:33:25	0:33:31	0:41:29

Y_train_dim (22345, 23335) Label Cardinality 3.0060 Training Samples/Label 0.9576
Y_test_dim (5587, 23335) Label Density 1.2882E-04

ตาราง 4.54 : เปรียบเทียบผลข้อมูล Linear 1-27932 ชุดทดสอบ สำหรับสัดส่วนการลดมิติ 1.0 ระหว่างแบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับ แบบจำลองตั้งต้น

Random Method	Linear 1-27932	Testing Result											
XMLC-PAO Ratio	org	XMLC-PAO-1 : 1.0						XMLC-PAO-2 : 1.0					
Adam Learning	0.003	0.0015	0.003	0.006	0.008	0.01	0.02	0.0015	0.003	0.006	0.008	0.01	0.02
mrr	0.0279	0.0249	0.0251	0.0233	0.0255	0.0244	0.0248	0.0254	0.0247	0.0219	0.0247	0.0253	0.0237
recall@5	1.2867	1.1993	1.2740	1.0165	1.1018	1.0915	0.9670	1.3817	1.1954	1.0388	1.2386	1.2216	1.0040
recall@10	2.1165	1.7378	1.9690	1.7645	1.6499	1.7325	1.6987	1.9926	1.6337	1.3219	1.5945	1.9094	1.3895
recall@30	4.8364	2.7776	2.6454	3.2276	3.0412	3.5073	3.2764	3.3278	2.8290	2.7660	3.0385	3.3822	2.5783
recall@50	6.4497	3.5325	3.4018	3.6753	4.2328	4.2245	4.0783	4.6927	3.5677	3.5106	4.0186	4.2457	3.3713
recall@100	9.4752	4.9704	5.4402	4.8041	6.1584	6.0371	5.6838	6.9219	5.7228	4.6592	5.6723	6.5127	5.2357
ndcg@5	1.1189	1.0953	1.1326	0.9756	1.0633	1.0076	0.9777	1.1508	1.0942	0.9467	1.1056	1.1259	0.9772
ndcg@10	1.4645	1.3227	1.4327	1.2900	1.2961	1.2751	1.2746	1.4181	1.2839	1.0831	1.2713	1.3879	1.1566
ndcg@30	2.2716	1.6571	1.6626	1.7318	1.7288	1.8282	1.7613	1.8223	1.6392	1.5246	1.7176	1.7983	1.5321
ndcg@50	2.6635	1.8538	1.8539	1.8750	2.0021	2.0372	1.9555	2.1507	1.8188	1.7064	1.9427	2.0138	1.7377
ndcg@100	3.3283	2.1618	2.2882	2.1270	2.3964	2.4119	2.3118	2.6092	2.2523	1.9763	2.3020	2.4905	2.1167
Process_data_time													
seconds	1589.9088	1645.8691	1616.3506	1651.1584	1665.4783	1651.1828	1650.3774	2035.4821	1992.6092	1994.1963	2010.8026	2028.2997	2518.3773
hrs	0:26:30	0:27:26	0:26:56	0:27:31	0:27:45	0:27:31	0:27:30	0:33:55	0:33:13	0:33:14	0:33:31	0:33:48	0:41:58

Y_train_dim	(22345, 23335)	Label Cardinality	3.0060	Training Samples/Label	0.9576
Y_test_dim	(5587, 23335)	Label Density	1.2882E-04		

ตารางผลการทดลองสำหรับข้อมูลชุดเรียนรู้

ผลการทดลองข้อมูลชุดเรียนรู้ สำหรับข้อมูล 5000 ตัวอย่าง

ตาราง 4.55 : เปรียบเทียบผลข้อมูล Linear 1-5000 ชุดเรียนรู้ สำหรับสัดส่วนการลดมิติ 0.5 ระหว่างแบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับ แบบจำลองดั้งต้น

Random Method	Linear 1-5000	Training Result											
XMLC-PAO Ratio	org	XMLC-PAO-1 : 0.5						XMLC-PAO-2 : 0.5					
Adam Learning	0.003	0.0015	0.003	0.006	0.008	0.01	0.02	0.0015	0.003	0.006	0.008	0.01	0.02
mrr	0.0530	0.0411	0.0477	**0.0552**	0.0520	0.0364	0.0479	0.0527	0.0430	0.0463	0.0533	0.0464	0.0415
recall@5	2.6688	1.6746	1.9336	**2.8005**	2.2632	1.3808	1.8240	2.4534	1.5914	1.7018	1.9828	1.7182	1.8013
recall@10	3.4988	2.1310	2.5445	**3.9640**	3.0937	1.9582	2.5980	3.1470	2.1658	2.2058	2.5875	2.2147	2.4705
recall@30	5.7748	3.1256	3.6262	**7.5978**	5.0876	3.1034	4.2588	4.3627	3.3328	4.8468	5.4789	3.6526	3.5906
recall@50	8.1266	4.6437	5.3395	**8.9454**	7.2957	3.9516	5.9322	5.5967	4.4665	6.1838	7.9660	4.8235	4.5493
recall@100	9.9998	9.5636	10.9129	**12.7374**	12.0585	8.8736	10.2298	11.7341	8.8113	11.6692	12.3358	10.1332	9.4982
ndcg@5	2.1715	1.5187	1.8303	**2.4656**	1.9797	1.2644	1.8091	2.2569	1.5626	1.7798	1.9537	1.7297	1.5992
ndcg@10	2.6272	1.7867	2.2106	**3.0960**	2.4114	1.5923	2.2115	2.6783	1.8972	2.1435	2.3495	2.0645	1.9485
ndcg@30	3.5683	2.3146	2.8532	**4.4699**	3.2934	2.2051	3.0395	3.4069	2.5056	3.1958	3.4400	2.7206	2.5551
ndcg@50	4.3063	2.7920	3.4437	**4.9853**	3.9954	2.5476	3.6750	3.9321	2.9207	3.7311	4.1783	3.1262	2.9564
ndcg@100	4.9463	4.0774	4.8705	**6.0790**	5.2696	3.8183	4.8520	5.4420	4.0623	5.0926	5.4296	4.4464	4.2693
Process_data_time													
seconds	286.0810	276.4867	**264.1901**	264.8814	266.7158	267.9199	269.1589	279.2522	280.8876	281.2558	282.3455	283.0533	284.3426
hrs	0:04:46	0:04:36	**0:04:24**	0:04:25	0:04:27	0:04:28	0:04:29	0:04:39	0:04:41	0:04:41	0:04:42	0:04:43	0:04:44

Y_train_dim	(4000, 6651)	Label Cardinality	4.4873	Training Samples/Label	0.6014
Y_test_dim	(1000, 6651)	Label Density	6.7467E-04		

ตาราง 4.56 : เปรียบเทียบผลข้อมูล Linear 1-5000 ชุดเรียนรู้ สำหรับสัดส่วนการลดมิติ 0.8 ระหว่างแบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับ แบบจำลองดั้งต้น

Random Method	Linear 1-5000	Training Result											
XMLC-PAO Ratio	org	XMLC-PAO-1 : 0.8						XMLC-PAO-2 : 0.8					
Adam Learning	0.003	0.0015	0.003	0.006	0.008	0.01	0.02	0.0015	0.003	0.006	0.008	0.01	0.02
mrr	0.0530	0.0405	0.0477	0.0351	0.0449	0.0484	0.0419	0.0543	0.0397	0.0480	0.0508	0.0513	**0.0565**
recall@5	2.6688	1.2701	2.0411	0.9775	1.5198	2.1671	1.5460	2.7290	1.7111	1.9802	1.9025	2.4369	**2.7701**
recall@10	3.4988	2.0181	2.8295	1.4889	1.9534	2.8001	1.9327	**3.6460**	2.1986	2.7666	2.5665	3.0611	3.5435
recall@30	5.7748	3.6969	4.5737	2.4998	2.9857	4.1751	2.8024	5.8216	3.2643	4.3705	4.2880	4.9129	**6.2496**
recall@50	8.1266	4.9627	6.1294	3.4412	4.4172	5.3770	5.6871	7.9147	4.0445	5.9984	5.8126	6.3455	**9.2006**
recall@100	9.9998	9.7073	11.0290	8.7331	10.8273	9.9543	10.5504	12.4902	9.4356	10.8396	9.6885	10.2433	**14.9377**
ndcg@5	2.1715	1.3390	1.8262	1.0836	1.6111	1.9627	1.5022	2.2334	1.4761	1.7891	1.7667	2.1136	**2.3794**
ndcg@10	2.6272	1.7347	2.2487	1.3583	1.9000	2.3226	1.7435	2.7014	1.7402	2.2092	2.1357	2.4880	**2.8067**
ndcg@30	3.5683	2.4422	3.0406	1.8539	2.5574	3.0803	2.2112	3.6628	2.2925	2.9953	2.9412	3.3742	**3.9178**
ndcg@50	4.3063	2.8905	3.5982	2.2076	3.1035	3.5574	3.0241	4.3594	2.6182	3.5739	3.4745	3.9145	**4.8051**
ndcg@100	4.9463	4.0974	4.8920	3.5489	4.6401	4.7691	4.3811	5.6431	3.9739	4.8714	4.6185	5.0139	**6.2692**
Process_data_time													
seconds	286.0810	**271.3268**	272.5074	274.0789	274.4366	276.4196	276.3945	288.9709	290.2965	291.2522	292.2474	293.8584	294.5012
hrs	0:04:46	**0:04:31**	0:04:33	0:04:34	0:04:34	0:04:36	0:04:36	0:04:49	0:04:50	0:04:51	0:04:52	0:04:54	0:04:55

Y_train_dim	(4000, 6651)	Label Cardinality	4.4873	Training Samples/Label	0.6014
Y_test_dim	(1000, 6651)	Label Density	6.7467E-04		

ตาราง 4.57 : เปรียบเทียบผลข้อมูล Linear 1-5000 ชุดเรียนรู้ สำหรับสัดส่วนการลดมิติ 1.0 ระหว่างแบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับ แบบจำลองตั้งต้น

Random Method	Linear 1-5000	Training Result											
XMLC-PAO Ratio	org	XMLC-PAO-1 : 1.0						XMLC-PAO-2 : 1.0					
Adam Learning	0.003	0.0015	0.003	0.006	0.008	0.01	0.02	0.0015	0.003	0.006	0.008	0.01	0.02
mrr	0.0530	0.0502	0.0510	0.0526	0.0465	0.0497	0.0534	0.0527	0.0410	0.0549	0.0460	0.0516	**0.0575**
recall@5	**2.6688**	2.2780	2.3951	1.7281	1.7483	2.1015	2.5545	2.2354	0.9664	2.6065	2.0715	1.8952	2.2368
recall@10	3.4988	3.1290	2.9652	2.9022	2.5550	2.8900	3.3975	2.9758	1.3757	3.5257	2.6006	2.6203	**3.7431**
recall@30	5.7748	5.0904	4.7193	5.7898	5.3326	4.9206	**5.8819**	4.9477	2.3475	5.4484	4.1981	3.7919	5.6726
recall@50	8.1266	6.8319	7.4090	**8.7288**	6.5476	6.7186	7.6022	7.2431	4.2856	6.9969	5.7738	6.1523	8.1432
recall@100	9.9998	11.7376	12.2826	**13.5124**	11.0132	11.9322	12.0413	11.4220	11.0900	11.7002	10.6519	10.7440	12.5039
ndcg@5	2.1715	2.0026	2.1022	1.8748	1.8075	1.8700	2.2067	1.9954	1.2390	**2.2777**	1.8053	1.9396	2.1285
ndcg@10	2.6272	2.4677	2.4611	2.4041	2.2626	2.2705	2.6378	2.4031	1.5040	2.7708	2.1188	2.3527	**2.8103**
ndcg@30	3.5683	3.3677	3.2977	3.4723	3.3749	3.1906	3.6598	3.2795	2.0936	3.7058	2.8074	3.0520	**3.7514**
ndcg@50	4.3063	4.0029	4.0916	4.3379	3.8851	3.8200	4.3197	4.0030	2.8370	4.3066	3.3594	3.7724	**4.5236**
ndcg@100	4.9463	5.3004	5.3591	5.5951	5.1198	5.1662	5.5496	5.1449	4.5036	5.5720	4.6398	5.0538	**5.7359**
Process_data_time													
seconds	286.0810	**278.7729**	281.3054	281.3162	282.5347	284.7627	284.4292	299.4775	300.5688	301.4037	302.7960	305.0587	304.0337
hrs	0:04:46	**0:04:39**	0:04:41	0:04:41	0:04:43	0:04:45	0:04:44	0:04:59	0:05:01	0:05:01	0:05:03	0:05:05	0:05:04

Y_train_dim	(4000, 6651)	**Label Cardinality**	4.4873	**Training Samples/Label**	0.6014
Y_test_dim	(1000, 6651)	**Label Density**	6.7467E-04		

ตาราง 4.58 : เปรียบเทียบผลข้อมูล Linear 201-5200 ชุดเรียนรู้ สำหรับสัดส่วนการลดมิติ 0.5 ระหว่างแบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับ แบบจำลองตั้งต้น

Random Method	Linear 201-5200	Training Result											
XMLC-PAO Ratio	org	XMLC-PAO-1 : 0.5						XMLC-PAO-2 : 0.5					
Adam Learning	0.003	0.0015	0.003	0.006	0.008	0.01	0.02	0.0015	0.003	0.006	0.008	0.01	0.02
mrr	0.0502	0.0443	0.0514	0.0396	0.0501	0.0249	0.0510	0.0481	0.0528	0.0256	0.0225	**0.0530**	0.0431
recall@5	1.7049	1.9077	2.2308	1.5287	2.3722	0.6208	1.8026	1.8951	2.2889	0.8829	0.5394	**2.7769**	1.4347
recall@10	3.5670	2.5038	3.1252	1.9939	3.0970	0.9060	2.7297	2.6035	3.2163	1.1055	0.6945	**3.7345**	2.4119
recall@30	6.2488	4.3742	5.5949	3.3348	4.6317	1.3988	5.0833	4.2955	6.1991	1.8564	1.1213	**7.0714**	3.9344
recall@50	8.1281	5.3548	7.2281	4.2967	6.3632	1.6139	7.5769	5.9534	8.0056	2.2020	1.3514	**9.7737**	5.2110
recall@100	10.9449	10.4435	12.4727	10.0031	9.7287	7.1040	11.0829	9.7268	12.0868	8.3801	4.2968	**14.1802**	10.8031
ndcg@5	1.8344	1.7453	1.9756	1.4244	1.9911	0.7119	1.9051	1.8326	2.1687	0.8467	0.5807	**2.3971**	1.5957
ndcg@10	2.6016	2.0726	2.4431	1.6886	2.3837	0.8891	2.4127	2.2030	2.6821	0.9966	0.6802	**2.9614**	2.1216
ndcg@30	3.6500	2.8265	3.4303	2.2780	3.1027	1.1636	3.4408	2.9639	3.8936	1.3569	0.9141	**4.1930**	2.8962
ndcg@50	4.2654	3.2139	4.0050	2.6526	3.6675	1.2984	4.1995	3.5571	4.5151	1.5306	1.0522	**4.9515**	3.3756
ndcg@100	5.0891	4.5080	5.3274	4.0460	4.6328	2.6674	5.2050	4.5358	5.5709	3.0314	1.9102	**6.0735**	4.7399
Process_data_time													
seconds	298.4902	274.6650	275.9238	277.4712	277.4948	279.5860	280.7333	287.7428	276.8370	277.8015	277.1003	**274.6248**	277.6804
hrs	0:04:58	0:04:35	0:04:36	0:04:37	0:04:37	0:04:40	0:04:41	0:04:48	0:04:37	0:04:38	0:04:37	**0:04:35**	0:04:38

Y_train_dim (4000, 6646) Label Cardinality 4.3305 Training Samples/Label 0.6019
Y_test_dim (1000, 6646) Label Density 6.5159E-04

ตาราง 4.59 : เปรียบเทียบผลข้อมูล Linear 201-5200 ชุดเรียนรู้ สำหรับสัดส่วนการลดมิติ 0.8 ระหว่างแบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับ แบบจำลองตั้งต้น

Random Method	Linear 201-5200	Training Result											
XMLC-PAO Ratio	org	XMLC-PAO-1 : 0.8						XMLC-PAO-2 : 0.8					
Adam Learning	0.003	0.0015	0.003	0.006	0.008	0.01	0.02	0.0015	0.003	0.006	0.008	0.01	0.02
mrr	0.0502	0.0514	0.0529	**0.0535**	0.0494	0.0321	0.0432	0.0523	0.0353	0.0526	0.0492	0.0279	0.0523
recall@5	1.7049	2.0372	2.5412	2.5303	2.3205	0.9607	1.9061	1.5876	1.0430	**2.6459**	2.2469	0.6483	2.1307
recall@10	3.5670	3.2734	3.3233	3.5330	3.0300	1.7033	2.5147	2.3668	1.3994	**3.6253**	2.9631	0.9578	3.2984
recall@30	6.2488	5.2978	6.0227	6.0852	5.2301	2.6052	4.1038	5.4735	2.0508	**6.7874**	5.1357	1.3879	5.6322
recall@50	8.1281	6.7194	7.9394	9.1903	6.6327	3.4327	4.8454	8.2157	2.6340	**9.3671**	6.5192	1.7450	8.0270
recall@100	10.9449	10.4760	11.6480	**14.8277**	9.8941	8.3667	8.6410	13.2779	7.6144	12.6804	10.3643	7.3908	11.4526
ndcg@5	1.8344	1.9643	2.2084	**2.2681**	2.0313	1.0400	1.7022	1.8564	1.1261	2.2101	2.0048	0.7219	2.0306
ndcg@10	2.6016	2.5392	2.6407	**2.8141**	2.4409	1.3819	2.0241	2.3079	1.3129	2.7129	2.4057	0.8909	2.5981
ndcg@30	3.6500	3.4169	3.7146	**3.9357**	3.3380	1.8676	2.6798	3.4709	1.6366	3.8936	3.2960	1.1342	3.6278
ndcg@50	4.2654	3.9839	4.3885	**4.7855**	3.8455	2.2109	2.9728	4.2655	1.8849	4.6425	3.7799	1.3068	4.3566
ndcg@100	5.0891	5.0524	5.4173	**6.1003**	4.8041	3.4763	4.0250	5.5535	3.1644	5.5924	4.8379	2.6917	5.3649
Process_data_time													
seconds	298.4902	283.5981	284.9692	285.9180	286.8207	288.2754	289.2524	282.9559	283.9396	**281.5435**	283.6406	283.0063	285.3268
hrs	0:04:58	0:04:44	0:04:45	0:04:46	0:04:47	0:04:48	0:04:49	0:04:43	0:04:44	**0:04:42**	0:04:44	0:04:43	0:04:45

Y_train_dim (4000, 6646) Label Cardinality 4.3305 Training Samples/Label 0.6019
Y_test_dim (1000, 6646) Label Density 6.5159E-04

ตาราง 4.60 : เปรียบเทียบผลข้อมูล Linear 201-5200 ชุดเรียนรู้ สำหรับสัดส่วนการลดมิติ 1.0 ระหว่างแบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับ แบบจำลองตั้งต้น

Random Method	Linear 201-5200	Training Result											
XMLC-PAO Ratio	org	XMLC-PAO-1 : 1.0						XMLC-PAO-2 : 1.0					
Adam Learning	0.003	0.0015	0.003	0.006	0.008	0.01	0.02	0.0015	0.003	0.006	0.008	0.01	0.02
mrr	0.0502	0.0485	0.0335	0.0373	0.0465	0.0529	0.0377	0.0499	0.0223	0.0525	0.0489	0.0453	**0.0537**
recall@5	1.7049	2.0929	1.1909	1.3768	1.7192	2.2379	1.3516	2.2200	0.4122	**2.8497**	2.0616	1.8232	2.3516
recall@10	3.5670	2.8378	1.5202	1.6338	2.5864	3.1008	1.7750	3.1614	0.5928	**3.8326**	2.8405	2.5464	3.2168
recall@30	6.2488	4.6084	2.4308	2.0814	4.3731	5.8273	2.7520	6.3355	0.9450	**6.6429**	5.9805	4.4632	5.0128
recall@50	8.1281	6.1273	3.0768	2.4661	5.1287	8.2955	3.7747	8.3013	1.2124	**8.8406**	8.5251	5.5031	6.3613
recall@100	10.9449	10.2527	8.4720	6.3789	8.9125	**13.9989**	8.9138	13.1000	6.2913	13.1473	12.3573	9.8986	10.0259
ndcg@5	1.8344	1.8947	1.2365	1.3849	1.7730	2.1238	1.3229	2.1027	0.5201	**2.3865**	2.0131	1.7233	2.0949
ndcg@10	2.6016	2.3265	1.4668	1.5370	2.2389	2.6075	1.5657	2.6281	0.6389	**2.9235**	2.4856	2.1281	2.5429
ndcg@30	3.6500	3.1605	1.9886	1.8131	3.0461	3.7125	2.0357	3.8517	0.8516	**4.0719**	3.6275	2.9623	3.3074
ndcg@50	4.2654	3.6671	2.2825	2.0139	3.3695	4.4738	2.3828	4.4836	0.9993	**4.7012**	4.4255	3.4018	3.7662
ndcg@100	5.0891	4.7933	3.6400	3.0913	4.3774	**5.8388**	3.6501	5.7035	2.2369	5.7815	5.4653	4.5369	4.8162
Process_data_time													
seconds	298.4902	291.8600	292.7168	294.1885	294.6789	296.2335	298.9938	**290.1914**	293.4103	294.9540	293.6254	295.2210	297.2030
hrs	0:04:58	0:04:52	0:04:53	0:04:54	0:04:55	0:04:56	0:04:59	**0:04:50**	0:04:53	0:04:55	0:04:54	0:04:55	0:04:57

Y_train_dim (4000, 6646) **Label Cardinality** 4.3305 **Training Samples/Label** 0.6019
Y_test_dim (1000, 6646) **Label Density** 6.5159E-04

ตาราง 4.61 : เปรียบเทียบผลข้อมูล Linear 801-5800 ชุดเรียนรู้ สำหรับสัดส่วนการลดมิติ 0.5 ระหว่างแบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับ แบบจำลองตั้งต้น

Random Method	Linear 801-5800	Training Result											
XMLC-PAO Ratio	org	XMLC-PAO-1 : 0.5						XMLC-PAO-2 : 0.5					
Adam Learning	0.003	0.0015	0.003	0.006	0.008	0.01	0.02	0.0015	0.003	0.006	0.008	0.01	0.02
mrr	0.0526	0.0504	0.0365	0.0522	0.0511	0.0476	0.0499	**0.0540**	0.0478	0.0523	0.0494	0.0539	0.0462
recall@5	**2.6878**	2.3996	1.4286	2.5505	1.9544	1.6587	2.3981	1.7898	2.0388	2.0730	2.3362	1.9636	2.1852
recall@10	3.4071	3.2146	1.8735	3.4473	2.9055	2.1567	2.9330	2.8860	2.5034	3.0302	2.9799	**3.4935**	2.9988
recall@30	**6.5548**	5.4760	2.9100	6.0658	4.6649	3.3397	4.0267	5.6218	4.3600	4.9370	4.7528	6.0978	6.2808
recall@50	**9.0975**	6.6166	3.9968	8.3730	6.4201	5.5428	4.8391	7.4182	5.7862	6.7372	6.3988	8.6852	8.7276
recall@100	11.1932	11.0130	8.6124	12.0860	11.3935	11.4162	8.0615	10.9725	10.3790	10.6876	11.7350	**13.8019**	12.3979
ndcg@5	**2.2791**	2.1551	1.3560	2.2560	2.0389	1.8084	2.1727	1.9523	1.8821	2.0794	2.0646	2.1169	1.9257
ndcg@10	2.6704	2.5831	1.5893	2.6863	2.4911	2.0941	2.4784	2.4432	2.1440	2.5445	2.4313	**2.7500**	2.3449
ndcg@30	3.7492	3.4552	2.0049	3.5965	3.2326	2.6819	3.0277	3.4650	2.8229	3.3952	3.1647	**3.7750**	3.4634
ndcg@50	4.4033	3.8893	2.3348	4.2300	3.7294	3.3495	3.3140	4.0032	3.2453	3.9572	3.6708	**4.4785**	4.1237
ndcg@100	5.0574	4.9330	3.4715	5.1824	4.8829	4.6959	4.1148	4.9237	4.3739	4.9193	4.9269	**5.6657**	5.0324
Process_data_time													
seconds	283.4980	278.3665	267.7825	267.5885	**265.9119**	268.1987	269.4404	298.2491	289.3050	290.1026	291.0396	292.9776	293.5673
hrs	0:04:43	0:04:38	0:04:28	0:04:28	**0:04:26**	0:04:28	0:04:29	0:04:58	0:04:49	0:04:50	0:04:51	0:04:53	0:04:54

Y_train_dim (4000, 6682) Label Cardinality 3.9830 Training Samples/Label 0.5986
Y_test_dim (1000, 6682) Label Density 5.9608E-04

ตาราง 4.62 : เปรียบเทียบผลข้อมูล Linear 801-5800 ชุดเรียนรู้ สำหรับสัดส่วนการลดมิติ 0.8 ระหว่างแบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับ แบบจำลองตั้งต้น

Random Method	Linear 801-5800	Training Result											
XMLC-PAO Ratio	org	XMLC-PAO-1 : 0.8						XMLC-PAO-2 : 0.8					
Adam Learning	0.003	0.0015	0.003	0.006	0.008	0.01	0.02	0.0015	0.003	0.006	0.008	0.01	0.02
mrr	0.0526	0.0515	0.0483	0.0532	0.0440	0.0504	0.0486	0.0394	0.0484	0.0530	0.0544	**0.0546**	0.0359
recall@5	**2.6878**	2.2122	1.9353	1.8400	1.5362	1.7768	2.0895	1.4081	1.8211	1.7616	2.6754	2.6646	1.3700
recall@10	3.4071	3.0664	2.4887	3.3512	2.3489	2.8950	2.9698	1.8590	2.9364	2.7429	3.6148	**3.7099**	1.7491
recall@30	6.5548	4.9711	4.0038	6.2444	4.0327	4.9354	5.2709	3.2201	4.5453	6.1449	6.8762	**6.9891**	2.8412
recall@50	9.0975	7.1065	5.2682	8.2747	7.0897	6.7728	7.2831	4.1282	5.7364	8.3089	**9.9631**	9.7377	3.8764
recall@100	11.1932	10.3370	9.7565	13.0138	11.5472	10.8374	11.2195	7.3922	8.5807	13.4773	**14.7792**	14.1080	8.5430
ndcg@5	2.2791	2.0893	1.9159	1.9925	1.6033	1.8971	1.9989	1.4237	1.7998	1.8945	2.3331	**2.3461**	1.3183
ndcg@10	2.6704	2.5468	2.2227	2.6621	1.9867	2.3919	2.4214	1.6610	2.2904	2.3552	2.7999	**2.8244**	1.5273
ndcg@30	3.7492	3.3850	2.9116	3.6776	2.6846	3.2014	3.3166	2.1730	2.9098	3.4611	3.9563	**3.9822**	1.9741
ndcg@50	4.4033	3.9884	3.3063	4.2844	3.5117	3.7002	3.8865	2.4545	3.2762	4.0734	**4.7456**	4.7070	2.2864
ndcg@100	5.0574	4.8573	4.3543	5.4341	4.6473	4.6793	4.8311	3.3257	4.0877	5.3140	**5.8820**	5.7649	3.4720
Process_data_time													
seconds	283.4980	**271.7933**	271.8287	273.0143	272.5572	276.1511	276.1874	298.3517	299.2540	300.3995	301.2700	302.5294	303.4267
hrs	0:04:43	**0:04:32**	0:04:32	0:04:33	0:04:33	0:04:36	0:04:36	0:04:58	0:04:59	0:05:00	0:05:01	0:05:03	0:05:03

Y_train_dim (4000, 6682) Label Cardinality 3.9830 Training Samples/Label 0.5986
Y_test_dim (1000, 6682) Label Density 5.9608E-04

ตาราง 4.63 : เปรียบเทียบผลข้อมูล Linear 801-5800 ชุดเรียนรู้ สำหรับสัดส่วนการลดมิติ 1.0 ระหว่างแบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับ แบบจำลองตั้งต้น

Random Method	Linear 801-5800	Training Result											
XMLC-PAO Ratio	org	XMLC-PAO-1 : 1.0						XMLC-PAO-2 : 1.0					
Adam Learning	0.003	0.0015	0.003	0.006	0.008	0.01	0.02	0.0015	0.003	0.006	0.008	0.01	0.02
mrr	0.0526	0.0523	0.0358	0.0443	0.0539	0.0530	0.0504	0.0401	0.0528	0.0541	0.0391	0.0459	**0.0547**
recall@5	**2.6878**	2.1628	1.4797	1.8414	2.3457	2.1746	2.2200	1.5703	1.8358	2.5597	1.4389	1.3560	2.3391
recall@10	3.4071	2.8366	1.8316	2.1731	3.1677	3.1116	3.1831	2.0228	3.3693	**3.4552**	1.8348	2.3033	3.1251
recall@30	**6.5548**	5.2757	2.9709	3.3397	5.5683	5.6258	5.8382	3.3733	5.8322	6.0610	2.7421	4.0150	6.0804
recall@50	**9.0975**	6.9709	3.5386	4.3570	7.8585	7.3729	7.8407	4.2154	7.5554	7.9850	5.9152	5.0808	8.6861
recall@100	11.1932	11.1610	6.0721	8.3134	12.1596	11.3384	12.4926	8.9772	11.7588	12.5138	10.8761	9.4082	**12.9797**
ndcg@5	2.2791	2.0537	1.3347	1.7780	2.1420	2.0971	2.0428	1.4989	1.9452	**2.2804**	1.4122	1.5577	2.1717
ndcg@10	2.6704	2.4296	1.5305	1.9857	2.5478	2.5677	2.5260	1.7439	2.5792	**2.7250**	1.6349	1.9699	2.6078
ndcg@30	**3.7492**	3.3088	1.9778	2.4765	3.4795	3.5107	3.4726	2.2548	3.5264	3.6507	2.0523	2.5948	3.6588
ndcg@50	**4.4033**	3.8180	2.1857	2.7837	4.1038	4.0361	4.0324	2.5258	4.0604	4.2014	2.9029	2.9243	4.3528
ndcg@100	5.0574	4.8301	2.9033	3.7668	5.1623	4.9994	5.1250	3.6646	5.0877	5.2925	4.1202	3.9850	**5.4492**
Process_data_time													
seconds	283.4980	**278.7431**	281.0906	279.1697	279.4847	281.9901	280.9967	307.7984	310.2095	309.8961	311.1111	311.4987	313.5110
hrs	0:04:43	**0:04:39**	0:04:41	0:04:39	0:04:39	0:04:42	0:04:41	0:05:08	0:05:10	0:05:10	0:05:11	0:05:11	0:05:14

Y_train_dim	(4000, 6682)	**Label Cardinality**	3.9830	**Training Samples/Label**	0.5986
Y_test_dim	(1000, 6682)	**Label Density**	5.9608E-04		

ตาราง 4.64 : เปรียบเทียบผลข้อมูล Linear 5001-10000 ชุดเรียนรู้ สำหรับสัดส่วนการลดมิติ 0.5 ระหว่างแบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับ แบบจำลองดั้งต้น

Random Method	Linear 5001-10000	Training Result											
XMLC-PAO Ratio	org	XMLC-PAO-1 : 0.5						XMLC-PAO-2 : 0.5					
Adam Learning	0.003	0.0015	0.003	0.006	0.008	0.01	0.02	0.0015	0.003	0.006	0.008	0.01	0.02
mrr	0.0406	**0.0431**	0.0380	0.0362	0.0403	0.0415	0.0417	0.0398	0.0417	0.0428	0.0404	0.0418	0.0399
recall@5	2.1079	2.4735	2.0453	1.7496	2.1905	**2.6242**	2.2708	2.0571	2.2688	2.2268	2.1387	2.2545	2.2991
recall@10	3.5460	4.0485	2.8400	2.5672	3.5943	**4.0504**	3.6353	3.0649	3.5433	3.8258	3.4562	3.6461	3.6201
recall@30	7.2993	7.1460	5.5467	6.7613	7.0178	7.3769	7.4844	6.5594	6.7364	**7.6444**	7.2398	7.3549	7.0919
recall@50	**10.3429**	9.6323	8.2443	9.2918	9.5870	10.2157	10.0402	8.8016	9.2913	9.7245	9.8534	9.3674	9.6025
recall@100	14.2135	14.2508	12.8056	13.3864	13.8878	**14.6160**	14.1765	13.5949	12.8703	13.9148	14.0535	14.1747	13.8622
ndcg@5	1.8651	2.1078	1.8135	1.6322	1.9171	**2.1406**	1.9853	1.8512	1.9736	1.9687	1.8834	1.9890	1.9415
ndcg@10	2.4452	2.7523	2.1804	2.0021	2.4745	**2.7680**	2.5259	2.2829	2.5121	2.5982	2.4028	2.5941	2.4805
ndcg@30	3.5751	3.7548	3.0283	3.2470	3.5191	**3.8129**	3.7553	3.3199	3.5145	3.7808	3.5276	3.7248	3.5929
ndcg@50	4.2804	4.3793	3.6656	3.8833	4.1516	**4.5136**	4.3948	3.8947	4.1569	4.3165	4.1572	4.2227	4.2093
ndcg@100	5.1902	5.3737	4.6303	4.7760	5.1248	**5.4839**	5.2953	4.9005	4.9851	5.2208	5.1383	5.2629	5.1557
Process_data_time													
seconds	276.5084	272.7186	263.1741	263.3456	263.2724	**261.5893**	262.5668	282.1414	273.1717	272.5722	273.8270	275.2396	276.1482
hrs	0:04:37	0:04:33	0:04:23	0:04:23	0:04:23	**0:04:22**	0:04:23	0:04:42	0:04:33	0:04:33	0:04:34	0:04:35	0:04:36

Y_train_dim	(4000, 6937)	Label Cardinality	3.1175	Training Samples/Label	0.5766
Y_test_dim	(1000, 6937)	Label Density	4.4940E-04		

ตาราง 4.65 : เปรียบเทียบผลข้อมูล Linear 5001-10000 ชุดเรียนรู้ สำหรับสัดส่วนการลดมิติ 0.8 ระหว่างแบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับ แบบจำลองดั้งต้น

Random Method	Linear 5001-10000	Training Result											
XMLC-PAO Ratio	org	XMLC-PAO-1 : 0.8						XMLC-PAO-2 : 0.8					
Adam Learning	0.003	0.0015	0.003	0.006	0.008	0.01	0.02	0.0015	0.003	0.006	0.008	0.01	0.02
mrr	0.0406	0.0409	0.0419	0.0428	0.0402	0.0410	0.0393	0.0379	0.0401	0.0411	**0.0428**	0.0426	0.0402
recall@5	2.1079	2.3670	2.4122	**2.4244**	2.2799	2.2459	2.2273	1.9946	2.3450	2.3079	2.3696	2.3569	2.1823
recall@10	3.5460	3.6944	3.5549	**3.8344**	3.5960	3.5791	3.6518	3.3914	3.6423	3.6690	3.6926	3.5665	3.7809
recall@30	7.2993	6.7228	6.7373	6.8959	**7.8178**	7.2143	6.4540	7.4064	7.1202	7.0332	7.4196	7.6585	6.7784
recall@50	10.3429	9.4901	8.6963	9.2384	**10.3517**	9.5340	9.0847	9.5495	9.8880	9.0069	10.0894	10.1083	9.1574
recall@100	14.2135	14.2809	12.7179	13.8056	**14.9816**	13.9347	13.1547	13.7819	14.7236	12.8591	13.7024	14.6202	13.3410
ndcg@5	1.8651	2.0092	**2.1226**	2.0828	1.9392	1.9527	1.8578	1.6909	1.9806	1.9872	2.0371	2.0317	1.8934
ndcg@10	2.4452	2.5926	2.6405	**2.6853**	2.4948	2.5074	2.4506	2.2480	2.5209	2.5403	2.5948	2.5480	2.5219
ndcg@30	3.5751	3.5573	3.6175	3.7246	3.7098	3.5800	3.3548	3.4173	3.5513	3.6232	3.7392	**3.7828**	3.4394
ndcg@50	4.2804	4.2414	4.1565	4.3048	4.3174	4.1500	3.9643	3.9417	4.2186	4.1336	**4.4299**	4.3834	4.0545
ndcg@100	5.1902	5.2371	5.0019	5.2987	5.3332	5.0802	4.8501	4.8575	5.2522	4.9714	5.2144	**5.3717**	4.9837
Process_data_time													
seconds	276.5084	**265.0896**	265.5832	267.0533	266.6951	269.1758	269.6548	282.4170	284.3386	285.5109	284.3518	285.7659	287.2942
hrs	0:04:37	**0:04:25**	0:04:26	0:04:27	0:04:27	0:04:29	0:04:30	0:04:42	0:04:44	0:04:46	0:04:44	0:04:46	0:04:47

Y_train_dim	(4000, 6937)	Label Cardinality	3.1175	Training Samples/Label	0.5766
Y_test_dim	(1000, 6937)	Label Density	4.4940E-04		

ตาราง 4.66 : เปรียบเทียบผลข้อมูล Linear 5001-10000 ชุดเรียนรู้ สำหรับสัดส่วนการลดมิติ 1.0 ระหว่างแบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับ แบบจำลองตั้งต้น

Random Method	Linear 5001-10000	Training Result											
XMLC-PAO Ratio	org	XMLC-PAO-1 : 1.0						XMLC-PAO-2 : 1.0					
Adam Learning	0.003	0.0015	0.003	0.006	0.008	0.01	0.02	0.0015	0.003	0.006	0.008	0.01	0.02
mrr	0.0406	0.0408	0.0412	0.0399	0.0423	0.0405	0.0395	0.0409	**0.0431**	0.0425	0.0420	0.0410	0.0410
recall@5	2.1079	2.3016	**2.4890**	2.0737	2.4431	2.2541	2.3101	2.1985	2.2218	2.2211	2.1689	2.2111	2.1815
recall@10	3.5460	3.6834	**4.0880**	3.4279	3.7644	3.4138	3.4357	3.7415	3.6510	3.5919	3.8711	3.7085	3.6461
recall@30	7.2993	**8.0718**	7.6059	6.8636	7.5157	7.1202	7.0046	6.7505	7.4752	7.6898	7.8737	7.4747	7.2856
recall@50	10.3429	**10.7310**	10.4700	9.4723	9.8618	9.8932	9.2466	9.2998	10.4835	9.8148	9.8817	9.4890	9.2953
recall@100	14.2135	**15.1573**	15.0257	13.8583	13.6486	14.1694	13.3414	13.4515	15.0243	14.2663	13.7992	13.9940	13.6249
ndcg@5	1.8651	1.9530	2.0402	1.9107	**2.0865**	1.9496	1.9514	1.9403	2.0182	1.9379	1.9550	1.9148	1.9165
ndcg@10	2.4452	2.5293	**2.6862**	2.4983	2.6575	2.4374	2.4286	2.5441	2.6217	2.5169	2.5928	2.5377	2.5044
ndcg@30	3.5751	3.7829	3.7853	3.5497	3.7657	3.5628	3.4647	3.4743	3.7457	3.7152	**3.8220**	3.6568	3.6195
ndcg@50	4.2804	4.4134	4.4739	4.2043	4.3469	4.2295	4.0565	4.0851	**4.4825**	4.2679	4.3447	4.1759	4.1362
ndcg@100	5.1902	5.4334	5.4550	5.1747	5.2241	5.1668	4.9930	5.0339	**5.4876**	5.2370	5.2348	5.1691	5.0802
Process_data_time													
seconds	276.5084	**271.6128**	273.7317	274.2819	274.4355	276.5168	275.5372	288.4194	291.7164	293.5335	293.1274	294.6327	295.4843
hrs	0:04:37	**0:04:32**	0:04:34	0:04:34	0:04:34	0:04:37	0:04:36	0:04:48	0:04:52	0:04:54	0:04:53	0:04:55	0:04:55

Y_train_dim	(4000, 6937)	Label Cardinality	3.1175	Training Samples/Label	0.5766
Y_test_dim	(1000, 6937)	Label Density	4.4940E-04		

ตาราง 4.67 : เปรียบเทียบผลข้อมูล Linear 10001-15000 ชุดเรียนรู้ สำหรับสัดส่วนการลดมิติ 0.5 ระหว่างแบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับ แบบจำลองตั้งต้น

Random Method	Linear 10001-15000	Training Result											
XMLC-PAO Ratio	org	XMLC-PAO-1 : 0.5						XMLC-PAO-2 : 0.5					
Adam Learning	0.003	0.0015	0.003	0.006	0.008	0.01	0.02	0.0015	0.003	0.006	0.008	0.01	0.02
mrr	0.0319	0.0312	**0.0345**	0.0317	0.0309	0.0306	0.0320	0.0306	0.0307	0.0317	0.0309	0.0318	0.0311
recall@5	1.5447	2.0793	2.0058	**2.3035**	1.6567	1.6945	2.1374	1.7448	1.6813	1.7096	2.0302	1.8273	1.7755
recall@10	3.1358	3.3724	3.0789	**3.4110**	2.9078	2.6625	3.1836	2.9062	2.9118	2.9394	3.3420	3.0285	2.9921
recall@30	6.2057	6.3752	**6.5152**	6.4626	5.7505	5.6426	6.1396	6.1184	5.9283	6.0355	6.2618	6.0474	6.0611
recall@50	8.6642	8.2926	**9.0432**	8.4776	7.6281	7.8321	8.4993	8.5281	7.9139	8.6055	8.2371	7.9765	8.0848
recall@100	**13.2364**	12.1430	13.1328	12.0152	11.3654	11.6151	12.0194	12.5138	12.4014	12.4495	12.4711	12.1849	11.8840
ndcg@5	1.3653	1.5726	**1.7210**	1.6774	1.4180	1.5093	1.6928	1.4333	1.3955	1.5350	1.5959	1.5047	1.4749
ndcg@10	1.9627	2.0699	2.1407	**2.1436**	1.9068	1.9049	2.1313	1.8986	1.9445	2.0185	2.1351	1.9795	1.9626
ndcg@30	2.9167	3.0035	**3.1852**	3.0558	2.7718	2.7777	3.0379	2.8619	2.8485	2.9618	3.0087	2.8926	2.9132
ndcg@50	3.5163	3.4938	**3.7766**	3.5580	3.2539	3.3052	3.5722	3.4305	3.3419	3.5766	3.4954	3.3658	3.4052
ndcg@100	4.4779	4.3529	**4.6161**	4.3277	4.0489	4.0842	4.3193	4.2391	4.2492	4.3854	4.3778	4.2502	4.1875
Process_data_time													
seconds	278.5305	259.2918	259.3344	**259.2489**	259.2999	261.0926	261.2234	299.6653	286.9303	287.9926	289.6439	291.2047	292.2794
hrs	0:04:39	0:04:19	0:04:19	**0:04:19**	0:04:19	0:04:21	0:04:21	0:05:00	0:04:47	0:04:48	0:04:50	0:04:51	0:04:52

Y_train_dim (4000, 7079) Label Cardinality 2.7965 Training Samples/Label 0.5651
Y_test_dim (1000, 7079) Label Density 3.9504E-04

ตาราง 4.68 : เปรียบเทียบผลข้อมูล Linear 10001-15000 ชุดเรียนรู้ สำหรับสัดส่วนการลดมิติ 0.8 ระหว่างแบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับ แบบจำลองตั้งต้น

Random Method	Linear 10001-15000	Training Result											
XMLC-PAO Ratio	org	XMLC-PAO-1 : 0.8						XMLC-PAO-2 : 0.8					
Adam Learning	0.003	0.0015	0.003	0.006	0.008	0.01	0.02	0.0015	0.003	0.006	0.008	0.01	0.02
mrr	0.0319	0.0314	0.0322	0.0322	0.0297	0.0331	0.0322	0.0296	0.0317	0.0320	**0.0331**	0.0275	0.0306
recall@5	1.5447	1.7674	2.1250	1.8650	1.8810	1.9724	1.8403	1.6161	1.9290	2.1439	**2.4220**	1.5746	1.9475
recall@10	3.1358	3.1375	3.4115	2.8620	2.7818	3.1066	3.2458	2.5110	3.2715	3.0776	**3.4987**	2.4892	3.2433
recall@30	6.2057	5.9201	6.6012	5.7192	6.2962	6.6928	6.3917	5.9569	5.9166	5.9539	**6.7128**	5.3929	6.0118
recall@50	8.6642	8.1506	8.5879	8.7241	8.5280	**8.9111**	8.4449	8.3958	8.1216	8.1762	8.5840	7.6720	8.0510
recall@100	**13.2364**	12.1914	12.4583	12.5268	11.6287	12.7228	12.2943	12.3827	11.5728	12.0705	12.5898	12.1627	11.8676
ndcg@5	1.3653	1.5036	1.6513	1.5408	1.4709	1.6952	1.5025	1.3225	1.5510	1.6828	**1.8386**	1.1874	1.4341
ndcg@10	1.9627	2.0490	2.1615	1.9841	1.8240	2.1416	2.0351	1.7017	2.0917	2.0685	**2.3145**	1.5740	1.9520
ndcg@30	2.9167	2.9334	3.1235	2.8416	2.8627	3.1980	2.9745	2.6984	2.9417	2.9730	**3.3111**	2.4886	2.8066
ndcg@50	3.5163	3.4709	3.6052	3.5101	3.4117	3.7437	3.4960	3.2828	3.4906	3.4985	**3.7672**	3.0437	3.2867
ndcg@100	4.4779	4.3408	4.4042	4.2945	4.1122	4.5132	4.3457	4.1285	4.2688	4.3199	**4.6374**	3.9744	4.0986
Process_data_time													
seconds	278.5305	**263.4875**	266.3922	266.5922	268.6146	269.1453	271.0775	297.1155	299.5397	300.4595	301.6624	302.4780	302.5011
hrs	0:04:39	**0:04:23**	0:04:26	0:04:27	0:04:29	0:04:29	0:04:31	0:04:57	0:05:00	0:05:00	0:05:02	0:05:02	0:05:03

Y_train_dim (4000, 7079) Label Cardinality 2.7965 Training Samples/Label 0.5651
Y_test_dim (1000, 7079) Label Density 3.9504E-04

ตาราง 4.69 : เปรียบเทียบผลข้อมูล Linear 10001-15000 ชุดเรียนรู้ สำหรับสัดส่วนการลดมิติ 1.0 ระหว่างแบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับ แบบจำลองตั้งต้น

Random Method	Linear 10001-15000	Training Result											
XMLC-PAO Ratio	org	XMLC-PAO-1 : 1.0						XMLC-PAO-2 : 1.0					
Adam Learning	0.003	0.0015	0.003	0.006	0.008	0.01	0.02	0.0015	0.003	0.006	0.008	0.01	0.02
mrr	0.0319	0.0298	0.0307	0.0324	0.0325	0.0308	0.0316	0.0319	0.0312	0.0312	0.0288	0.0315	**0.0330**
recall@5	1.5447	1.8224	2.0324	**2.4888**	2.1331	1.9939	1.7822	1.6964	1.9845	2.0490	1.5494	1.6906	1.6661
recall@10	3.1358	2.5416	3.2644	**3.6392**	3.0494	3.1113	2.9513	2.9056	2.9743	3.2267	2.7730	3.0540	2.7626
recall@30	6.2057	5.4305	6.0858	6.1306	6.3478	5.9900	5.7942	6.2439	5.9365	**6.5518**	5.7341	6.0228	6.2929
recall@50	8.6642	7.4436	7.8984	8.0144	8.3136	8.1440	8.0001	8.7768	7.7696	**8.7900**	7.7302	8.2804	8.7529
recall@100	**13.2364**	11.4923	12.0039	11.9888	11.8519	12.2562	12.6714	12.7398	11.4818	12.3101	10.5266	12.3976	12.2007
ndcg@5	1.3653	1.4973	1.6074	**1.8154**	1.6358	1.5689	1.4902	1.4279	1.6604	1.5696	1.3406	1.4315	1.4793
ndcg@10	1.9627	1.8465	2.0753	**2.2739**	2.0282	2.0044	1.9884	1.9189	2.0753	2.0384	1.8559	1.9499	1.9515
ndcg@30	2.9167	2.7265	2.9524	**3.0650**	3.0172	2.8760	2.8581	2.9199	3.0033	3.0508	2.7269	2.8942	2.9772
ndcg@50	3.5163	3.2305	3.3911	3.5623	3.5080	3.3788	3.3882	3.5161	3.4652	**3.5866**	3.2041	3.4446	3.5661
ndcg@100	**4.4779**	4.1020	4.2640	4.3786	4.2769	4.2595	4.3302	4.3496	4.2609	4.3447	3.8100	4.2919	4.2928
Process_data_time													
seconds	278.5305	272.2162	**271.5172**	273.0762	274.5566	276.2453	278.0829	306.7906	307.0073	308.6738	309.9557	309.4086	312.5458
hrs	0:04:39	0:04:32	**0:04:32**	0:04:33	0:04:35	0:04:36	0:04:38	0:05:07	0:05:07	0:05:09	0:05:10	0:05:09	0:05:13

Y_train_dim	(4000, 7079)	Label Cardinality	2.7965	Training Samples/Label	0.5651
Y_test_dim	(1000, 7079)	Label Density	3.9504E-04		

ตาราง 4.70 : เปรียบเทียบผลข้อมูล Linear 12201-17200 ชุดเรียนรู้ สำหรับสัดส่วนการลดมิติ 0.5 ระหว่างแบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับ แบบจำลองตั้งต้น

Random Method	Linear 12201-17200	Training Result											
XMLC-PAO Ratio	org	XMLC-PAO-1 : 0.5						XMLC-PAO-2 : 0.5					
Adam Learning	0.003	0.0015	0.003	0.006	0.008	0.01	0.02	0.0015	0.003	0.006	0.008	0.01	0.02
mrr	0.0303	0.0294	0.0300	0.0308	0.0292	**0.0327**	0.0311	0.0301	0.0307	0.0303	0.0301	0.0296	0.0292
recall@5	2.0598	2.0019	2.0230	2.1814	**2.2729**	2.1609	1.9692	1.9388	2.0781	1.9256	1.9394	1.8371	1.8221
recall@10	3.1640	3.1312	2.9051	2.9939	2.9723	3.1042	2.7883	2.8849	3.0651	**3.2981**	3.0332	2.8709	2.9179
recall@30	6.0844	5.7296	5.6417	5.6347	5.3722	5.7920	6.0309	5.9166	5.4747	5.9247	5.7976	**6.2728**	5.6584
recall@50	**8.4493**	7.5353	7.4674	7.7062	7.1119	8.0091	7.6250	7.9405	7.4208	8.1952	7.4806	8.0259	7.9814
recall@100	**12.6393**	11.6580	11.3953	11.9048	9.7214	11.4866	10.8455	11.4483	11.4849	11.8583	11.0831	11.2126	11.4214
ndcg@5	1.4885	1.4691	1.6254	1.6720	1.6517	**1.7854**	1.5652	1.5261	1.5908	1.4751	1.5649	1.4591	1.3908
ndcg@10	1.9295	1.9171	1.9817	2.0135	1.9456	**2.1835**	1.9072	1.9095	1.9995	2.0018	2.0034	1.8761	1.8180
ndcg@30	2.8168	2.7030	2.8159	2.8177	2.6859	**2.9873**	2.8401	2.7832	2.7408	2.7735	2.8466	2.8586	2.6414
ndcg@50	3.3823	3.1402	3.2533	3.3127	3.1139	**3.5060**	3.2157	3.2567	3.2054	3.3062	3.2627	3.2687	3.2020
ndcg@100	**4.2692**	3.9907	4.0515	4.1628	3.7118	4.2286	3.9043	3.9972	4.0340	4.0820	4.0142	3.9219	3.9295
Process_data_time													
seconds	291.1387	**270.0433**	270.7243	271.5634	272.5843	273.8946	274.9241	292.5255	279.9617	280.8622	281.3905	282.0465	283.6917
hrs	0:04:51	**0:04:30**	0:04:31	0:04:32	0:04:33	0:04:34	0:04:35	0:04:53	0:04:40	0:04:41	0:04:41	0:04:42	0:04:44

Y_train_dim	(4000, 7088)	Label Cardinality	2.6660	Training Samples/Label	0.5643
Y_test_dim	(1000, 7088)	Label Density	3.7613E-04		

ตาราง 4.71 : เปรียบเทียบผลข้อมูล Linear 12201-17200 ชุดเรียนรู้ สำหรับสัดส่วนการลดมิติ 0.8 ระหว่างแบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับ แบบจำลองตั้งต้น

Random Method	Linear 12201-17200	Training Result											
XMLC-PAO Ratio	org	XMLC-PAO-1 : 0.8						XMLC-PAO-2 : 0.8					
Adam Learning	0.003	0.0015	0.003	0.006	0.008	0.01	0.02	0.0015	0.003	0.006	0.008	0.01	0.02
mrr	0.0303	0.0301	0.0272	0.0257	0.0304	0.0297	0.0295	0.0275	0.0300	0.0304	**0.0308**	0.0290	0.0287
recall@5	2.0598	1.9910	1.8154	1.6302	2.0034	1.5723	2.1504	1.8173	2.0547	**2.2387**	1.9209	1.7270	1.8140
recall@10	3.1640	3.1260	3.1037	2.5155	3.0391	2.8885	3.0535	2.8617	3.1359	**3.2373**	2.9745	2.8263	2.9143
recall@30	6.0844	5.6104	5.9682	5.3902	6.2000	5.6462	5.7283	5.5251	5.7993	**6.4106**	6.0165	5.9528	5.8350
recall@50	**8.4493**	7.4907	8.1039	7.6703	8.3813	7.3677	8.0665	7.5255	7.8183	8.3537	7.8469	7.4721	7.4809
recall@100	**12.6393**	11.2742	11.7840	10.8431	11.9476	10.9983	12.0037	11.4401	11.4245	11.9384	10.7186	11.0144	11.5456
ndcg@5	1.4885	1.5361	1.3809	1.3031	1.6027	1.3491	**1.6931**	1.3404	1.5480	1.6837	1.6081	1.4393	1.3917
ndcg@10	1.9295	1.9760	1.8658	1.6536	2.0235	1.8577	2.0661	1.7548	1.9811	**2.1060**	2.0344	1.8812	1.8312
ndcg@30	2.8168	2.7235	2.7321	2.4950	2.9731	2.6655	2.8852	2.5591	2.7658	**3.0784**	2.9227	2.7982	2.6974
ndcg@50	3.3823	3.1740	3.2417	3.0305	3.5004	3.0866	3.4259	3.0295	3.2686	**3.5514**	3.3650	3.1966	3.1010
ndcg@100	4.2692	3.9704	3.9920	3.7027	4.2559	3.8394	4.2397	3.8338	4.0311	**4.2934**	3.9630	3.9461	3.9463
Process_data_time													
seconds	291.1387	**277.7499**	278.3283	280.5989	279.9142	281.9208	282.1760	289.9873	290.3195	296.7917	292.3971	293.4979	292.9487
hrs	0:04:51	**0:04:38**	0:04:38	0:04:41	0:04:40	0:04:42	0:04:42	0:04:50	0:04:50	0:04:57	0:04:52	0:04:53	0:04:53

Y_train_dim	(4000, 7088)	Label Cardinality	2.6660	Training Samples/Label	0.5643
Y_test_dim	(1000, 7088)	Label Density	3.7613E-04		

ตาราง 4.72 : เปรียบเทียบผลข้อมูล Linear 12201-17200 ชุดเรียนรู้ สำหรับสัดส่วนการลดมิติ 1.0 ระหว่างแบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับ แบบจำลองตั้งต้น

Random Method	Linear 12201-17200	Training Result											
XMLC-PAO Ratio	org	XMLC-PAO-1 : 1.0						XMLC-PAO-2 : 1.0					
Adam Learning	0.003	0.0015	0.003	0.006	0.008	0.01	0.02	0.0015	0.003	0.006	0.008	0.01	0.02
mrr	0.0303	0.0300	0.0315	0.0299	0.0301	0.0288	0.0283	0.0288	0.0296	**0.0321**	0.0313	0.0304	0.0305
recall@5	2.0598	2.1346	1.9401	1.7421	1.9185	1.9246	1.6097	1.4847	1.9081	2.0358	1.9362	2.1181	**2.2200**
recall@10	3.1640	**3.3002**	3.0988	2.8043	3.2489	2.8896	2.7276	2.5977	2.9894	3.0801	3.0873	3.2880	3.2432
recall@30	6.0844	6.1728	6.1071	5.7813	5.8322	5.9245	5.8744	5.9686	5.9844	**6.3820**	5.6468	5.9073	5.4962
recall@50	**8.4493**	7.5693	8.2771	7.6206	7.6316	8.0651	7.5966	7.9811	8.1884	8.1168	7.7798	7.7112	7.4978
recall@100	**12.6393**	11.4321	11.8562	11.3922	10.9244	12.0784	11.4364	11.6158	11.6300	11.4399	11.4508	11.4902	10.6086
ndcg@5	1.4885	1.6798	1.5111	1.3916	1.4861	1.4685	1.3833	1.2573	1.4334	1.6644	1.5600	1.5633	**1.7398**
ndcg@10	1.9295	2.1356	1.9878	1.8076	2.0166	1.8643	1.8295	1.6958	1.8766	2.0804	2.0160	2.0277	**2.1522**
ndcg@30	2.8168	2.9751	2.8825	2.6881	2.7898	2.7819	2.7203	2.6503	2.7605	**3.0160**	2.7798	2.8349	2.8906
ndcg@50	3.3823	3.3164	3.4001	3.1571	3.2159	3.2800	3.1371	3.1289	3.2715	**3.4375**	3.2849	3.2740	3.3837
ndcg@100	**4.2692**	4.1231	4.1571	3.9351	3.9006	4.1014	3.9301	3.9006	4.0126	4.1228	4.0535	4.0471	4.0234
Process_data_time													
seconds	291.1387	**283.6918**	286.4412	285.7947	287.4845	289.7898	289.4181	296.4826	298.6041	300.3685	300.7847	300.2447	301.4797
hrs	0:04:51	**0:04:44**	0:04:46	0:04:46	0:04:47	0:04:50	0:04:49	0:04:56	0:04:59	0:05:00	0:05:01	0:05:00	0:05:01

Y_train_dim (4000, 7088) **Label Cardinality** 2.6660 **Training Samples/Label** 0.5643
Y_test_dim (1000, 7088) **Label Density** 3.7613E-04

ตาราง 4.73 : เปรียบเทียบผลข้อมูล Linear 15001-20000 ชุดเรียนรู้ สำหรับสัดส่วนการลดมิติ 0.5 ระหว่างแบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับ แบบจำลองตั้งต้น

Random Method	Linear 15001-20000	Training Result											
XMLC-PAO Ratio	org	XMLC-PAO-1 : 0.5						XMLC-PAO-2 : 0.5					
Adam Learning	0.003	0.0015	0.003	0.006	0.008	0.01	0.02	0.0015	0.003	0.006	0.008	0.01	0.02
mrr	0.0274	0.0267	0.0258	**0.0282**	0.0275	0.0266	0.0264	0.0248	0.0280	0.0260	0.0269	0.0264	0.0238
recall@5	1.5291	1.7305	1.6099	1.6072	1.5385	**1.7842**	1.5336	1.5302	1.7583	1.5971	1.7461	1.4705	1.6134
recall@10	2.7535	2.6680	2.3878	2.5898	2.4589	2.6369	2.4689	2.3715	**2.7729**	2.5798	2.5565	2.4189	2.6832
recall@30	5.5551	5.0306	4.7136	5.2163	5.1697	4.9504	4.9514	5.0160	5.4629	5.2959	**5.5981**	5.0030	5.4454
recall@50	7.6912	7.0216	6.8122	7.4486	6.8135	6.6180	6.8990	7.2578	7.4750	7.1758	**7.7194**	6.8509	6.8402
recall@100	**11.6212**	10.7400	10.1011	10.7947	9.9595	10.7306	10.6109	10.5455	11.0868	10.5456	10.7816	10.1805	9.5586
ndcg@5	1.2465	1.4029	1.3215	1.3745	1.3666	1.3492	1.3148	1.2090	**1.4079**	1.2834	1.4024	1.2583	1.1817
ndcg@10	1.6977	1.7693	1.6621	1.7648	1.7217	1.6819	1.7054	1.5416	**1.8036**	1.6731	1.7322	1.6389	1.6083
ndcg@30	2.5314	2.4790	2.3340	2.5308	2.5355	2.3525	2.4362	2.3243	2.6014	2.4798	**2.6124**	2.3959	2.4188
ndcg@50	3.0605	2.9589	2.8491	3.0526	2.9240	2.7674	2.8873	2.8464	3.0761	2.9318	**3.1170**	2.8281	2.7551
ndcg@100	**3.8919**	3.7205	3.5541	3.7454	3.5917	3.6039	3.6521	3.5578	3.8104	3.6292	3.7480	3.5503	3.3167
Process_data_time													
seconds	279.5561	274.7197	**262.6380**	264.1856	264.3792	265.5783	267.0645	286.0428	274.8751	275.3273	276.4055	277.9537	277.9149
hrs	0:04:40	0:04:35	**0:04:23**	0:04:24	0:04:24	0:04:26	0:04:27	0:04:46	0:04:35	0:04:35	0:04:36	0:04:38	0:04:38

Y_train_dim (4000, 7228) **Label Cardinality** 2.6535 **Training Samples/Label** 0.5534
Y_test_dim (1000, 7228) **Label Density** 3.6711E-04

ตาราง 4.74 : เปรียบเทียบผลข้อมูล Linear 15001-20000 ชุดเรียนรู้ สำหรับสัดส่วนการลดมิติ 0.8 ระหว่างแบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับ แบบจำลองตั้งต้น

Random Method	Linear 15001-20000	Training Result											
XMLC-PAO Ratio	org	XMLC-PAO-1 : 0.8						XMLC-PAO-2 : 0.8					
Adam Learning	0.003	0.0015	0.003	0.006	0.008	0.01	0.02	0.0015	0.003	0.006	0.008	0.01	0.02
mrr	0.0274	0.0275	0.0251	0.0272	0.0254	0.0285	0.0249	**0.0286**	0.0258	0.0266	0.0270	0.0257	0.0271
recall@5	1.5291	1.3359	1.4522	1.5941	1.3949	1.6054	1.5367	1.6146	1.4746	1.5981	**1.8149**	1.7646	1.6480
recall@10	**2.7535**	2.6151	2.4167	2.4865	2.5318	2.4910	2.4042	2.6581	2.3691	2.5248	2.7442	2.5223	2.5615
recall@30	5.5551	**5.7014**	4.8284	4.9385	5.1312	5.4919	5.1764	5.4830	4.9678	5.5315	5.4268	5.3664	5.3545
recall@50	**7.6912**	7.4696	6.7109	7.3250	7.0515	7.5067	7.1571	7.3111	7.0613	7.6094	7.4356	7.3429	7.2157
recall@100	**11.6212**	10.8518	10.0142	10.8019	10.2648	11.0767	10.3191	10.8329	10.5394	10.7437	10.6171	10.9120	10.7293
ndcg@5	1.2465	1.1552	1.1714	1.3760	1.2040	**1.4177**	1.2029	1.4099	1.2409	1.3397	1.3873	1.3674	1.3694
ndcg@10	1.6977	1.6306	1.5516	1.7518	1.6430	1.7803	1.5551	**1.8343**	1.6144	1.7190	1.7558	1.6756	1.7360
ndcg@30	2.5314	2.4909	2.2689	2.4671	2.4109	**2.6559**	2.3613	2.6484	2.4001	2.6117	2.5339	2.5171	2.5468
ndcg@50	3.0605	2.9171	2.7274	3.0301	2.8789	**3.1317**	2.8209	3.0750	2.8875	3.1091	3.0145	2.9949	2.9966
ndcg@100	**3.8919**	3.6075	3.4382	3.7412	3.5496	3.8735	3.4747	3.7900	3.6109	3.7764	3.6935	3.7203	3.7355
Process_data_time													
seconds	279.5561	**268.9874**	269.5131	270.9056	271.2401	272.7897	272.5271	284.5734	285.5161	287.6207	287.0519	290.1875	289.2449
hrs	0:04:40	**0:04:29**	0:04:30	0:04:31	0:04:31	0:04:33	0:04:33	0:04:45	0:04:46	0:04:48	0:04:47	0:04:50	0:04:49

Y_train_dim (4000, 7228) **Label Cardinality** 2.6535 **Training Samples/Label** 0.5534
Y_test_dim (1000, 7228) **Label Density** 3.6711E-04

ตาราง 4.75 : เปรียบเทียบผลข้อมูล Linear 15001-20000 ชุดเรียนรู้ สำหรับสัดส่วนการลดมิติ 1.0 ระหว่างแบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับ แบบจำลองตั้งต้น

Random Method	Linear 15001-20000	Training Result											
XMLC-PAO Ratio	org	XMLC-PAO-1 : 1.0						XMLC-PAO-2 : 1.0					
Adam Learning	0.003	0.0015	0.003	0.006	0.008	0.01	0.02	0.0015	0.003	0.006	0.008	0.01	0.02
mrr	0.0274	0.0267	0.0261	0.0254	0.0265	0.0237	0.0273	0.0274	0.0252	0.0254	0.0261	0.0264	0.0244
recall@5	1.5291	1.5934	1.5844	1.3443	1.5503	1.4848	1.8416	1.5861	1.4205	1.4659	1.6820	1.7820	1.5432
recall@10	2.7535	2.5016	2.5889	2.1752	2.1927	2.3701	2.6885	2.5690	2.4103	2.2302	2.4845	2.6133	2.3050
recall@30	5.5551	5.1794	5.5689	5.1452	5.2411	4.9041	5.4715	5.4771	4.6938	4.9651	5.2428	5.1312	4.7254
recall@50	7.6912	7.4299	7.4434	6.9918	7.2795	6.7167	6.8143	7.4950	6.5598	7.0612	7.1297	6.9381	6.7719
recall@100	11.6212	10.9293	10.7952	10.1047	10.0847	10.6206	10.7762	11.0895	10.5120	10.6527	10.6424	10.4426	10.0576
ndcg@5	1.2465	1.2880	1.2461	1.1452	1.2392	1.1944	1.3739	1.3288	1.1885	1.2209	1.2968	1.3433	1.1775
ndcg@10	1.6977	1.6460	1.6569	1.4693	1.5018	1.5506	1.7042	1.7034	1.5921	1.5269	1.6370	1.6858	1.4908
ndcg@30	2.5314	2.4580	2.5198	2.3122	2.3931	2.3008	2.5147	2.5156	2.2751	2.3318	2.4340	2.4166	2.2415
ndcg@50	3.0605	2.9707	2.9626	2.7707	2.8742	2.7316	2.8339	2.9865	2.7210	2.8226	2.8942	2.8580	2.7202
ndcg@100	3.8919	3.6819	3.6714	3.4344	3.4773	3.5135	3.6302	3.7319	3.5326	3.5654	3.6395	3.5921	3.3927
Process_data_time													
seconds	279.5561	274.9714	277.1300	277.0900	277.7818	279.8261	279.3560	294.0639	294.8051	294.5351	296.0705	298.0651	298.1227
hrs	0:04:40	0:04:35	0:04:37	0:04:37	0:04:38	0:04:40	0:04:39	0:04:54	0:04:55	0:04:55	0:04:56	0:04:58	0:04:58

Y_train_dim	(4000, 7228)	Label Cardinality	2.6535	Training Samples/Label	0.5534
Y_test_dim	(1000, 7228)	Label Density	3.6711E-04		

ตาราง 4.76 : เปรียบเทียบผลข้อมูล Linear 20001-25000 ชุดเรียนรู้ สำหรับสัดส่วนการลดมิติ 0.5 ระหว่างแบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับ แบบจำลองตั้งต้น

Random Method	Linear 20001-25000	Training Result											
XMLC-PAO Ratio	org	XMLC-PAO-1 : 0.5						XMLC-PAO-2 : 0.5					
Adam Learning	0.003	0.0015	0.003	0.006	0.008	0.01	0.02	0.0015	0.003	0.006	0.008	0.01	0.02
mrr	0.0276	**0.0278**	0.0265	0.0261	0.0265	0.0269	0.0278	0.0260	0.0276	0.0263	0.0264	0.0267	0.0275
recall@5	1.6442	1.7939	1.6094	1.7918	1.6600	1.8816	1.5933	1.5438	1.8669	1.6937	1.8952	**2.0587**	1.7057
recall@10	2.8933	2.9449	2.9507	3.0578	2.7712	3.0430	2.7487	2.7041	**3.1423**	2.8793	3.0003	2.9435	2.9280
recall@30	**6.3731**	5.6632	5.6834	5.9515	5.9742	5.9186	5.7915	5.6741	6.2326	5.8624	6.0906	6.0487	5.5230
recall@50	**8.7147**	7.7241	7.2271	8.0611	8.2154	7.6697	7.8338	7.5527	8.0104	7.8608	7.5203	7.8756	7.6759
recall@100	**13.0176**	10.8880	10.7227	11.8054	11.5983	11.0218	11.4540	11.2071	12.2435	11.2694	10.9917	11.2417	10.9644
ndcg@5	1.2441	1.3548	1.2898	1.2797	1.2543	1.3597	1.2841	1.1903	1.3714	1.2677	1.3607	**1.5059**	1.3216
ndcg@10	1.7146	1.7846	1.8052	1.7564	1.6852	1.8285	1.7213	1.6337	1.8299	1.7120	1.7889	**1.8593**	1.7681
ndcg@30	2.6786	2.5538	2.5585	2.6119	2.5832	2.6786	2.5553	2.4524	2.7158	2.5447	2.6656	**2.7223**	2.5050
ndcg@50	**3.2163**	3.0188	2.9413	3.1193	3.0877	3.0958	3.0345	2.8832	3.1248	3.0033	2.9999	3.1502	2.9766
ndcg@100	**4.0771**	3.6626	3.6493	3.8796	3.7835	3.7668	3.7574	3.6021	3.9499	3.6892	3.6960	3.8452	3.6429
Process_data_time													
seconds	291.5006	**271.8910**	273.4393	274.2250	274.8275	278.9125	279.9659	288.3216	277.1843	277.6070	279.3432	279.0132	280.8723
hrs	0:04:52	**0:04:32**	0:04:33	0:04:34	0:04:35	0:04:39	0:04:40	0:04:48	0:04:37	0:04:38	0:04:39	0:04:39	0:04:41

Y_train_dim (4000, 6931) **Label Cardinality** 2.4700 **Training Samples/Label** 0.5771
Y_test_dim (1000, 6931) **Label Density** 3.5637E-04

ตาราง 4.77 : เปรียบเทียบผลข้อมูล Linear 20001-25000 ชุดเรียนรู้ สำหรับสัดส่วนการลดมิติ 0.8 ระหว่างแบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับ แบบจำลองตั้งต้น

Random Method	Linear 20001-25000	Training Result											
XMLC-PAO Ratio	org	XMLC-PAO-1 : 0.8						XMLC-PAO-2 : 0.8					
Adam Learning	0.003	0.0015	0.003	0.006	0.008	0.01	0.02	0.0015	0.003	0.006	0.008	0.01	0.02
mrr	0.0276	0.0261	0.0262	**0.0281**	0.0267	0.0261	0.0258	0.0268	0.0244	0.0256	0.0270	0.0244	0.0274
recall@5	1.6442	1.6420	1.4609	**2.0312**	1.8125	1.5920	1.7311	1.9710	1.5762	1.6502	1.6522	1.8036	1.7014
recall@10	2.8933	2.7489	2.5612	**3.3167**	2.7020	2.6822	2.5852	2.8789	2.4596	2.7838	2.7152	2.7531	2.9396
recall@30	**6.3731**	5.6243	5.3974	6.1636	5.9019	5.2022	5.4534	5.5823	5.1391	5.8895	5.5656	5.0184	6.0651
recall@50	**8.7147**	7.7010	7.4363	7.9432	7.6397	7.4199	7.4760	7.7777	7.5193	7.7052	7.5590	7.1267	7.5870
recall@100	**13.0176**	11.1806	11.6129	11.2422	10.8113	11.1602	10.8686	12.1882	11.5345	11.3861	11.0323	10.4367	11.1262
ndcg@5	1.2441	1.2807	1.1856	**1.5220**	1.3477	1.2429	1.3077	1.4251	1.2042	1.2201	1.2963	1.2900	1.2978
ndcg@10	1.7146	1.7162	1.6138	**1.9912**	1.7070	1.6703	1.6253	1.7629	1.5634	1.6478	1.7059	1.6573	1.7459
ndcg@30	2.6786	2.5433	2.4142	**2.7968**	2.6190	2.4003	2.4223	2.5416	2.3464	2.5587	2.5261	2.3431	2.5990
ndcg@50	**3.2163**	3.0108	2.8647	3.2076	3.0354	2.8996	2.8647	3.0321	2.8706	2.9784	2.9627	2.8430	2.9447
ndcg@100	**4.0771**	3.6902	3.6860	3.8935	3.6803	3.6493	3.5483	3.8882	3.6660	3.7244	3.6631	3.5268	3.6750
Process_data_time													
seconds	291.5006	**280.5030**	280.8394	281.2258	282.3380	282.8808	285.6473	285.2144	288.2474	286.3446	289.1967	288.7589	290.9288
hrs	0:04:52	**0:04:41**	0:04:41	0:04:41	0:04:42	0:04:43	0:04:46	0:04:45	0:04:48	0:04:46	0:04:49	0:04:49	0:04:51

Y_train_dim (4000, 6931) **Label Cardinality** 2.4700 **Training Samples/Label** 0.5771
Y_test_dim (1000, 6931) **Label Density** 3.5637E-04

ตาราง 4.78 : เปรียบเทียบผลข้อมูล Linear 20001-25000 ชุดเรียนรู้ สำหรับสัดส่วนการลดมิติ 1.0 ระหว่างแบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับ แบบจำลองตั้งต้น

Random Method	Linear 20001-25000	Training Result											
XMLC-PAO Ratio	org	XMLC-PAO-1 : 1.0						XMLC-PAO-2 : 1.0					
Adam Learning	0.003	0.0015	0.003	0.006	0.008	0.01	0.02	0.0015	0.003	0.006	0.008	0.01	0.02
mrr	0.0276	0.0258	**0.0280**	0.0249	0.0239	0.0264	0.0236	0.0272	0.0261	0.0259	0.0273	0.0274	0.0255
recall@5	1.6442	1.6761	1.8086	1.8486	1.6072	1.5535	1.5981	1.8505	1.6405	1.6054	1.6325	**1.9661**	1.7467
recall@10	2.8933	2.7302	2.8969	3.1557	2.9565	2.5566	2.8219	2.9893	2.8492	2.8719	2.9183	**3.3460**	2.9849
recall@30	6.3731	5.3634	**6.4392**	5.5584	5.8673	6.0536	5.2115	5.9654	5.5047	5.6199	6.3321	5.8286	5.6851
recall@50	**8.7147**	7.3602	8.5136	7.3690	7.8119	8.1748	7.1057	7.9887	7.5829	7.7311	8.3000	7.8682	7.9891
recall@100	**13.0176**	10.9862	11.6433	11.1592	11.2711	11.8781	10.1174	11.5799	11.2696	10.8209	11.8279	11.4133	11.3165
ndcg@5	1.2441	1.3074	1.3847	1.3103	1.1245	1.1937	1.2003	1.3375	1.2466	1.2591	1.2681	**1.4283**	1.2955
ndcg@10	1.7146	1.7201	1.7860	1.8263	1.6496	1.5963	1.6606	1.7368	1.6887	1.7325	1.7477	**1.9171**	1.7434
ndcg@30	2.6786	2.4671	**2.7683**	2.5467	2.4677	2.5786	2.3499	2.5914	2.4547	2.5119	2.6698	2.6409	2.4767
ndcg@50	3.2163	2.9087	**3.2472**	2.9868	2.9120	3.0676	2.7791	3.0649	2.9101	2.9979	3.1311	3.0968	3.0052
ndcg@100	**4.0771**	3.6324	3.8819	3.7331	3.5971	3.8000	3.4009	3.7760	3.6430	3.6272	3.8633	3.8038	3.6675
Process_data_time													
seconds	291.5006	**286.2104**	288.3949	289.3650	290.0340	290.7418	291.1257	293.4787	295.6047	295.1565	297.2512	296.8167	299.7163
hrs	0:04:52	**0:04:46**	0:04:48	0:04:49	0:04:50	0:04:51	0:04:51	0:04:53	0:04:56	0:04:55	0:04:57	0:04:57	0:05:00

Y_train_dim	(4000, 6931)	Label Cardinality	2.4700	Training Samples/Label	0.5771
Y_test_dim	(1000, 6931)	Label Density	3.5637E-04		

ตาราง 4.79 : เปรียบเทียบผลข้อมูล Linear 25001-27932 (2933 ตัวอย่าง) ชุดเรียนรู้ สำหรับสัดส่วนการลดมิติ 0.5 ระหว่างแบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับแบบจำลองตั้งต้น

Random Method	Linear 25001-27932	Training Result											
XMLC-PAO Ratio	org	XMLC-PAO-1 : 0.5						XMLC-PAO-2 : 0.5					
Adam Learning	0.003	0.0015	0.003	0.006	0.008	0.01	0.02	0.0015	0.003	0.006	0.008	0.01	0.02
mrr	0.0232	0.0247	0.0246	0.0250	0.0241	0.0240	0.0241	0.0244	0.0241	0.0235	0.0245	0.0249	**0.0253**
recall@5	1.5081	1.5248	1.5374	1.7764	1.7125	1.6491	1.5431	1.7269	1.6293	1.5588	1.4518	**1.8034**	1.6926
recall@10	2.2876	2.2826	2.5182	2.6220	2.7055	2.7559	2.4773	2.8165	2.3099	2.4686	**2.8319**	2.5273	2.8127
recall@30	4.8863	5.6638	5.5890	5.5907	5.5654	5.4942	5.3986	**5.9378**	5.5300	5.2328	5.6725	5.3333	5.8113
recall@50	7.1488	8.2371	7.6739	7.7633	7.6783	7.5905	7.4774	**8.4224**	7.9287	7.4587	7.7222	7.3409	8.2168
recall@100	11.7963	12.6810	12.1414	**13.2321**	11.8489	12.4788	12.0373	12.4142	12.2728	11.8738	12.2846	12.0897	12.6560
ndcg@5	1.1830	1.2110	1.2326	**1.3247**	1.2809	1.2355	1.1778	1.2366	1.2452	1.2041	1.1583	1.3116	1.3247
ndcg@10	1.5054	1.5504	1.6258	1.6639	1.6751	1.6816	1.5677	1.6588	1.5260	1.5801	1.6535	1.6120	**1.7618**
ndcg@30	2.2854	2.5043	2.5012	2.5327	2.4914	2.4747	2.4128	2.5256	2.4627	2.4044	2.5049	2.4579	**2.6148**
ndcg@50	2.8106	3.0896	2.9856	3.0307	2.9688	2.9681	2.8893	3.1001	3.0189	2.9207	2.9692	2.9123	**3.1794**
ndcg@100	3.7383	3.9685	3.8833	**4.1077**	3.7650	3.9704	3.8055	3.9278	3.8885	3.8324	3.8742	3.8492	4.0768
Process_data_time													
seconds	170.1603	**155.9716**	156.3979	156.8777	157.6429	158.4585	159.3564	159.3495	160.2767	160.7963	161.2509	162.4565	162.9972
hrs	0:02:50	**0:02:36**	0:02:36	0:02:37	0:02:38	0:02:38	0:02:39	0:02:39	0:02:40	0:02:41	0:02:41	0:02:42	0:02:43

Y_train_dim	(2346, 4792)	Label Cardinality	2.4224	Training Samples/Label	0.4896
Y_test_dim	(587, 4792)	Label Density	5.0551E-04		

ตาราง 4.80 : เปรียบเทียบผลข้อมูล Linear 25001-27932 (2933 ตัวอย่าง) ชุดเรียนรู้ สำหรับสัดส่วนการลดมิติ 0.8 ระหว่างแบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับแบบจำลองตั้งต้น

Random Method	Linear 25001-27932	Training Result											
XMLC-PAO Ratio	org	XMLC-PAO-1 : 0.8						XMLC-PAO-2 : 0.8					
Adam Learning	0.003	0.0015	0.003	0.006	0.008	0.01	0.02	0.0015	0.003	0.006	0.008	0.01	0.02
mrr	0.0232	0.0253	0.0243	0.0239	0.0265	0.0249	0.0248	0.0262	0.0255	0.0257	0.0249	**0.0278**	0.0244
recall@5	1.5081	1.8115	1.5573	1.5618	1.7657	1.7087	1.6492	1.7889	1.6830	**1.9838**	1.7177	1.6287	1.5434
recall@10	2.2876	2.8479	2.3878	2.6254	**2.9787**	2.7754	2.8521	2.6525	2.5421	2.6541	2.5755	2.6262	2.5621
recall@30	4.8863	5.3238	5.6749	5.2472	5.8641	5.6616	5.5046	**5.8768**	5.6513	4.9257	5.8425	5.7332	5.5463
recall@50	7.1488	8.2650	7.8828	7.6145	8.2373	7.9723	7.9979	**8.3480**	7.9991	7.8080	8.1787	7.8429	8.1533
recall@100	11.7963	**13.0681**	12.8732	12.2214	12.4859	12.3307	12.6541	12.8591	13.0366	12.1498	12.7395	12.6766	12.0551
ndcg@5	1.1830	1.3283	1.2036	1.2068	1.3399	1.2924	1.2707	1.3918	1.3082	**1.4027**	1.2569	1.3402	1.2140
ndcg@10	1.5054	1.7446	1.5266	1.6171	**1.8064**	1.6990	1.7342	1.7333	1.6646	1.6970	1.5981	1.7465	1.6170
ndcg@30	2.2854	2.4690	2.4626	2.4024	2.6397	2.4992	2.5288	**2.6518**	2.5480	2.4084	2.5196	2.6438	2.4962
ndcg@50	2.8106	3.1365	2.9776	2.9519	3.2032	3.0393	3.0908	**3.2156**	3.0963	3.0559	3.0618	3.1433	3.1020
ndcg@100	3.7383	4.1025	3.9727	3.8772	4.0910	3.9158	4.0472	**4.1168**	4.1124	3.9230	3.9746	4.0993	3.9031
Process_data_time													
seconds	170.1603	**160.1624**	160.9981	162.4114	162.5840	163.1437	163.8967	164.2075	165.6152	165.8609	166.6519	167.5537	169.1035
hrs	0:02:50	**0:02:40**	0:02:41	0:02:42	0:02:43	0:02:43	0:02:44	0:02:44	0:02:46	0:02:46	0:02:47	0:02:48	0:02:49

Y_train_dim	(2346, 4792)	Label Cardinality	2.4224	Training Samples/Label	0.4896
Y_test_dim	(587, 4792)	Label Density	5.0551E-04		

ตาราง 4.81 : เปรียบเทียบผลข้อมูล Linear 25001-27932 (2933 ตัวอย่าง) ชุดเรียนรู้ สำหรับสัดส่วนการลดมิติ 1.0 ระหว่างแบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับแบบจำลองตั้งต้น

Random Method	Linear 25001-27932	Training Result											
XMLC-PAO Ratio	org	XMLC-PAO-1 : 1.0						XMLC-PAO-2 : 1.0					
Adam Learning	0.003	0.0015	0.003	0.006	0.008	0.01	0.02	0.0015	0.003	0.006	0.008	0.01	0.02
mrr	0.0232	0.0258	0.0235	0.0254	0.0246	**0.0272**	0.0244	0.0257	0.0272	0.0251	0.0247	0.0237	0.0241
recall@5	1.5081	1.3287	1.6262	1.9017	1.6259	**1.9544**	1.5313	1.8748	1.8123	1.6286	1.5850	1.7772	1.6847
recall@10	2.2876	2.4809	2.4692	2.7706	2.5844	**2.7712**	2.6362	2.6537	2.6737	2.4827	2.5983	2.6963	2.6105
recall@30	4.8863	5.6022	5.7223	5.6907	5.2886	**5.9605**	5.9356	5.3172	5.7012	5.7801	5.7905	5.8558	5.3918
recall@50	7.1488	8.0289	8.1375	8.1887	7.5817	**8.2969**	8.2495	7.6233	8.0938	7.8976	7.7777	7.8797	7.4217
recall@100	11.7963	12.3828	12.5264	12.8623	11.8869	**13.1359**	12.1656	11.9232	12.9168	12.2698	12.0099	12.5171	12.3821
ndcg@5	1.1830	1.1644	1.1810	1.3801	1.2530	**1.5523**	1.2002	1.4139	1.4115	1.2398	1.2271	1.3294	1.2915
ndcg@10	1.5054	1.6107	1.5097	1.7215	1.6405	**1.9002**	1.6288	1.7214	1.7807	1.5731	1.6400	1.6891	1.6733
ndcg@30	2.2854	2.5230	2.4320	2.5695	2.4323	**2.8166**	2.5193	2.5033	2.6599	2.5111	2.5562	2.5737	2.5033
ndcg@50	2.8106	3.0866	2.9959	3.1786	2.9603	**3.3631**	3.0753	3.0569	3.2061	3.0324	3.0361	3.0535	2.9776
ndcg@100	3.7383	3.9670	3.8836	4.0851	3.8239	**4.3450**	3.8985	3.9341	4.1777	3.9109	3.9050	3.9681	3.9630
Process_data_time													
seconds	170.1603	**165.1016**	166.5484	166.4967	167.2005	169.3134	168.7673	169.6006	170.7693	171.4299	172.7543	173.1211	175.9085
hrs	0:02:50	**0:02:45**	0:02:47	0:02:46	0:02:47	0:02:49	0:02:49	0:02:50	0:02:51	0:02:51	0:02:53	0:02:53	0:02:56

Y_train_dim	(2346, 4792)	**Label Cardinality**	2.4224	**Training Samples/Label**	0.4896
Y_test_dim	(587, 4792)	**Label Density**	5.0551E-04		

ตาราง 4.82 : เปรียบเทียบผลข้อมูล Random seed-42 ชุดเรียนรู้ สำหรับสัดส่วนการลดมิติ 0.5 ระหว่างแบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับ แบบจำลองดั้งต้น

Random Method	Random seed-42	Training Result											
XMLC-PAO Ratio	org	XMLC-PAO-1 : 0.5						XMLC-PAO-2 : 0.5					
Adam Learning	0.003	0.0015	0.003	0.006	0.008	0.01	0.02	0.0015	0.003	0.006	0.008	0.01	0.02
mrr	0.0284	0.0252	0.0245	0.0212	0.0245	0.0221	0.0280	**0.0293**	0.0250	0.0278	0.0265	0.0279	0.0280
recall@5	1.1560	1.0239	0.9103	0.9484	1.0879	0.8182	1.1933	**1.3312**	1.1041	1.2189	1.1106	1.1553	1.1610
recall@10	**2.1318**	1.5022	1.5468	1.3890	1.6657	1.3181	1.8977	2.0637	1.6002	1.9787	2.0154	1.8833	1.8181
recall@30	**4.7302**	3.1051	3.2239	2.4311	3.0274	2.2489	4.0221	4.3210	3.0874	4.0695	4.2021	3.9906	4.1417
recall@50	**6.5501**	4.1737	4.2875	2.9227	3.7459	2.7303	5.9057	6.0033	4.0563	5.8575	5.7337	5.7566	5.7834
recall@100	9.0190	6.9283	7.4512	6.0290	6.2727	5.9830	8.6131	8.5109	6.8946	9.0538	8.7688	8.5203	**9.1587**
ndcg@5	1.0778	0.9822	0.9243	0.8464	0.9766	0.8104	1.0965	**1.1768**	1.0009	1.1284	0.9935	1.0653	1.0713
ndcg@10	1.4735	1.2050	1.1989	1.0476	1.2378	1.0308	1.4118	**1.5154**	1.2424	1.4506	1.3813	1.3949	1.3634
ndcg@30	**2.2637**	1.7574	1.7724	1.3920	1.6958	1.3505	2.0984	2.2080	1.7469	2.0991	2.0712	2.0927	2.0458
ndcg@50	**2.7196**	2.0631	2.0859	1.5530	1.9032	1.4956	2.5745	2.6342	2.0191	2.5204	2.4775	2.5543	2.4536
ndcg@100	**3.3305**	2.7006	2.7778	2.2240	2.4766	2.1961	3.2164	3.2071	2.6515	3.2286	3.1684	3.2013	3.2259
Process_data_time													
seconds	269.3646	281.5127	268.7470	**268.2193**	269.3863	272.0557	272.7078	304.1104	305.9943	302.6955	304.8159	303.5382	305.3572
hrs	**0:04:29**	0:04:35	0:04:35	0:04:35	0:04:36	0:04:36	0:04:38	0:05:17	0:05:18	0:05:19	0:05:32	0:05:24	0:05:27

Y_train_dim (4000, 8468) Label Cardinality 3.1473 Training Samples/Label 0.4724
Y_test_dim (1000, 8468) Label Density 3.7166E-04

ตาราง 4.83 : เปรียบเทียบผลข้อมูล Random seed-42 ชุดเรียนรู้ สำหรับสัดส่วนการลดมิติ 0.8 ระหว่างแบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับ แบบจำลองดั้งต้น

Random Method	Random seed-42	Training Result											
XMLC-PAO Ratio	org	XMLC-PAO-1 : 0.8						XMLC-PAO-2 : 0.8					
Adam Learning	0.003	0.0015	0.003	0.006	0.008	0.01	0.02	0.0015	0.003	0.006	0.008	0.01	0.02
mrr	0.0284	0.0279	0.0239	0.0277	0.0283	**0.0297**	0.0263	0.0287	0.0279	0.0277	0.0286	0.0138	0.0281
recall@5	1.1560	1.2910	1.0177	1.3058	1.3630	1.3049	1.0638	**1.3854**	1.2385	1.1701	1.2025	0.4912	1.2086
recall@10	2.1318	2.0995	1.5310	1.9578	**2.3721**	2.3551	1.6524	2.0880	2.0472	1.7197	2.1671	0.8225	2.0039
recall@30	**4.7302**	4.0934	2.9673	4.0578	4.0816	4.4782	2.9724	4.2432	4.4275	3.8789	4.6044	1.1800	4.1099
recall@50	**6.5501**	5.2547	3.5974	5.4793	6.1161	6.1021	3.9618	5.6458	6.0172	5.3685	5.9876	1.4666	5.3457
recall@100	9.0190	7.5680	6.3473	8.1076	9.2024	8.6448	5.9793	8.6858	8.5959	8.0555	**9.4695**	5.1059	8.1507
ndcg@5	1.0778	1.1305	0.9306	1.1493	1.1667	1.1384	1.0293	**1.1884**	1.0810	1.0671	1.1176	0.4655	1.0895
ndcg@10	1.4735	1.4767	1.1668	1.4295	**1.5916**	1.5623	1.3083	1.4912	1.4134	1.3197	1.5105	0.6102	1.4180
ndcg@30	2.2637	2.1358	1.6484	2.1059	2.1565	**2.2692**	1.8236	2.1677	2.1694	1.9901	2.2592	0.7604	2.0867
ndcg@50	**2.7196**	2.4549	1.8449	2.4736	2.6448	2.6865	2.0972	2.5359	2.5841	2.3923	2.6202	0.8625	2.4292
ndcg@100	3.3305	3.0252	2.4550	3.0930	3.3366	3.2913	2.5690	3.2284	3.2140	3.0052	**3.3744**	1.6356	3.0814
Process_data_time													
seconds	**269.3646**	275.3721	275.0418	274.5494	276.1146	275.5278	278.2145	316.9397	317.6689	319.1348	331.6014	323.7464	326.9723
hrs	**0:04:29**	0:04:35	0:04:35	0:04:35	0:04:36	0:04:36	0:04:38	0:05:17	0:05:18	0:05:19	0:05:32	0:05:24	0:05:27

Y_train_dim (4000, 8468) Label Cardinality 3.1473 Training Samples/Label 0.4724
Y_test_dim (1000, 8468) Label Density 3.7166E-04

ตาราง 4.84 : เปรียบเทียบผลข้อมูล Random seed-42 ชุดเรียนรู้ สำหรับสัดส่วนการลดมิติ 1.0 ระหว่างแบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับ แบบจำลองตั้งต้น

Random Method	Random seed-42	Training Result											
XMLC-PAO Ratio	org	XMLC-PAO-1 : 1.0						XMLC-PAO-2 : 1.0					
Adam Learning	0.003	0.0015	0.003	0.006	0.008	0.01	0.02	0.0015	0.003	0.006	0.008	0.01	0.02
mrr	0.0284	0.0254	0.0274	0.0231	0.0265	0.0238	**0.0286**	0.0243	0.0283	0.0254	0.0226	0.0256	0.0275
recall@5	1.1560	1.3159	1.2206	0.9550	1.3365	0.9381	1.2816	1.1796	**1.4679**	1.0334	1.0092	1.1006	1.1614
recall@10	2.1318	2.1034	1.9835	1.4271	1.9040	1.4823	**2.1725**	1.8261	2.0987	1.7116	1.4732	1.7248	1.8788
recall@30	4.7302	4.5743	3.7463	2.8444	3.6896	3.2030	**4.9340**	3.4008	4.2636	3.3553	2.7550	3.1731	4.1707
recall@50	**6.5501**	5.9265	4.6835	3.5730	5.4616	4.2242	6.5080	4.1514	5.8504	4.4811	3.3735	4.3323	5.8218
recall@100	**9.0190**	8.5591	7.8282	5.8731	8.4793	7.3302	9.0063	7.2075	8.3022	7.0593	7.0657	7.2809	8.3022
ndcg@5	1.0778	1.0626	1.1069	0.8960	1.1461	0.8966	1.1548	0.9695	**1.3087**	0.9613	0.8986	1.0008	1.0606
ndcg@10	1.4735	1.3910	1.4220	1.1088	1.4100	1.1422	1.5193	1.2329	**1.6272**	1.2545	1.0988	1.2907	1.3703
ndcg@30	2.2637	2.1467	1.9962	1.5495	2.0368	1.7097	**2.3689**	1.7496	2.2988	1.8076	1.5320	1.7978	2.0971
ndcg@50	2.7196	2.5036	2.2611	1.7626	2.4697	1.9926	**2.7869**	1.9666	2.7109	2.1295	1.7218	2.1158	2.5248
ndcg@100	3.3305	3.1180	2.9712	2.3061	3.1519	2.6831	**3.3681**	2.6280	3.2776	2.7371	2.5156	2.7649	3.0908
Process_data_time													
seconds	**269.3646**	279.5240	281.1356	279.8193	280.3088	282.1792	283.5612	312.2808	300.3509	301.8863	303.5976	302.4060	304.4920
hrs	**0:04:29**	0:04:35	0:04:35	0:04:35	0:04:36	0:04:36	0:04:38	0:05:17	0:05:18	0:05:19	0:05:32	0:05:24	0:05:27

Y_train_dim	(4000, 8468)	**Label Cardinality**	3.1473	**Training Samples/Label**	0.4724
Y_test_dim	(1000, 8468)	**Label Density**	3.7166E-04		

ตาราง 4.85 : เปรียบเทียบผลข้อมูล Random seed-101 ชุดเรียนรู้ สำหรับสัดส่วนการลดมิติ 0.5 ระหว่างแบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับ แบบจำลองตั้งต้น

Random Method	Random seed-101	Training Result											
XMLC-PAO Ratio	org	XMLC-PAO-1 : 0.5						XMLC-PAO-2 : 0.5					
Adam Learning	0.003	0.0015	0.003	0.006	0.008	0.01	0.02	0.0015	0.003	0.006	0.008	0.01	0.02
mrr	0.0321	0.0301	0.0307	0.0325	0.0293	**0.0329**	0.0311	0.0309	0.0310	0.0312	0.0312	0.0311	0.0315
recall@5	1.4795	1.4254	**1.5848**	1.5457	1.4957	1.4617	1.5816	1.4936	1.5181	1.5410	1.4098	1.5558	1.4398
recall@10	2.2952	2.0632	2.3144	**2.3436**	2.1329	2.1875	2.3169	2.1665	2.1312	2.3092	2.0776	2.3008	2.3212
recall@30	**4.9859**	4.3744	4.7588	4.4379	3.8648	4.1651	4.5467	4.3559	4.4428	4.9492	4.4391	4.7034	4.2817
recall@50	**6.7364**	6.3581	6.4454	6.2449	5.2757	5.5764	6.1168	5.9831	6.2259	6.5350	6.5244	6.5611	5.7346
recall@100	**10.9576**	9.0512	8.8752	9.6840	8.6997	8.5900	9.1671	8.8640	9.4319	10.0512	9.7763	9.7147	8.1271
ndcg@5	1.3717	1.3370	1.3900	**1.4758**	1.3604	1.4104	1.4335	1.3833	1.4015	1.3994	1.3526	1.3935	1.3906
ndcg@10	1.7071	1.6270	1.7040	**1.8273**	1.6358	1.7131	1.7269	1.6850	1.6746	1.7070	1.6417	1.7147	1.7277
ndcg@30	**2.5056**	2.3432	2.4458	2.4846	2.1964	2.3454	2.4136	2.3649	2.3739	2.4810	2.3676	2.4685	2.3404
ndcg@50	**2.9653**	2.8259	2.8858	2.9227	2.5578	2.7229	2.7873	2.7754	2.7906	2.8842	2.8819	2.9152	2.7108
ndcg@100	**3.8442**	3.4143	3.4156	3.6325	3.3226	3.3800	3.4379	3.3984	3.4862	3.6282	3.5860	3.5935	3.2612
Process_data_time													
seconds	284.6269	279.2733	**266.3802**	267.2192	268.2536	269.5964	270.6657	280.7266	276.4966	277.9987	278.8824	279.9648	279.7141
hrs	0:04:45	**0:04:33**	0:04:33	0:04:35	0:04:36	0:04:37	0:04:38	0:04:49	0:04:50	0:04:52	0:04:52	0:04:52	0:04:52

Y_train_dim	(4000, 8422)	Label Cardinality	2.9568	Training Samples/Label	0.4749
Y_test_dim	(1000, 8422)	Label Density	3.5107E-04		

ตาราง 4.86 : เปรียบเทียบผลข้อมูล Random seed-101 ชุดเรียนรู้ สำหรับสัดส่วนการลดมิติ 0.8 ระหว่างแบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับ แบบจำลองตั้งต้น

Random Method	Random seed-101	Training Result											
XMLC-PAO Ratio	org	XMLC-PAO-1 : 0.8						XMLC-PAO-2 : 0.8					
Adam Learning	0.003	0.0015	0.003	0.006	0.008	0.01	0.02	0.0015	0.003	0.006	0.008	0.01	0.02
mrr	0.0321	0.0303	0.0306	**0.0325**	0.0319	0.0302	0.0310	0.0308	0.0314	0.0298	0.0310	0.0259	0.0285
recall@5	1.4795	1.3081	1.5342	**1.6317**	1.6212	1.4415	1.6204	1.4469	1.5224	1.3086	1.4718	1.3345	1.3789
recall@10	2.2952	2.0537	2.3295	2.3906	2.3693	2.0518	2.2571	2.4022	**2.4665**	1.9331	2.1415	1.6846	2.1174
recall@30	**4.9859**	4.3296	4.7516	4.6701	4.9659	4.0754	4.3358	4.7524	4.8322	4.1473	4.4479	2.7950	3.7046
recall@50	6.7364	6.1562	6.4888	6.2231	**6.9420**	5.9090	5.9382	6.7028	6.6260	5.7500	6.0998	3.5898	4.6414
recall@100	**10.9576**	9.4130	9.6914	9.7424	10.1881	8.7452	9.2175	10.3519	9.5045	8.7258	9.1444	6.6778	7.6525
ndcg@5	1.3717	1.2305	1.3821	**1.4577**	1.4461	1.3535	1.4400	1.3416	1.3897	1.2920	1.3672	1.2512	1.2937
ndcg@10	1.7071	1.5500	1.7070	1.7712	1.7578	1.6317	1.7102	1.7097	**1.7738**	1.5807	1.6509	1.4052	1.5920
ndcg@30	2.5056	2.2595	2.4144	2.4624	**2.5579**	2.2685	2.3482	2.4314	2.4932	2.2452	2.3378	1.7759	2.0858
ndcg@50	2.9653	2.7355	2.8581	2.8436	**3.0351**	2.7354	2.7490	2.9044	2.9345	2.6521	2.7577	1.9767	2.3311
ndcg@100	**3.8442**	3.4650	3.5484	3.5636	3.7266	3.3309	3.4342	3.6807	3.5727	3.2992	3.4154	2.6236	2.9794
Process_data_time													
seconds	284.6269	**273.1241**	273.4103	275.0971	275.9106	277.1411	277.6546	289.4682	290.4975	291.7195	292.1839	291.7422	292.0908
hrs	0:04:45	**0:04:33**	0:04:33	0:04:35	0:04:36	0:04:37	0:04:38	0:04:49	0:04:50	0:04:52	0:04:52	0:04:52	0:04:52

Y_train_dim	(4000, 8422)	Label Cardinality	2.9568	Training Samples/Label	0.4749
Y_test_dim	(1000, 8422)	Label Density	3.5107E-04		

ตาราง 4.87 : เปรียบเทียบผลข้อมูล Random seed-101 ชุดเรียนรู้ สำหรับสัดส่วนการลดมิติ 1.0 ระหว่างแบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับ แบบจำลองตั้งต้น

Random Method	Random seed-101	Training Result											
XMLC-PAO Ratio	org	XMLC-PAO-1 : 1.0						XMLC-PAO-2 : 1.0					
Adam Learning	0.003	0.0015	0.003	0.006	0.008	0.01	0.02	0.0015	0.003	0.006	0.008	0.01	0.02
mrr	0.0321	0.0315	0.0302	0.0312	0.0306	0.0314	0.0307	0.0317	0.0314	**0.0322**	0.0305	0.0317	0.0302
recall@5	1.4795	1.4486	1.5189	1.4955	1.6079	1.6177	1.4650	1.4061	1.5494	1.4861	**1.6531**	1.6122	1.4775
recall@10	2.2952	2.3500	2.1691	2.3209	2.3935	2.2680	2.0330	2.2062	**2.4903**	2.4351	2.4185	2.3933	1.9940
recall@30	4.9859	**5.0875**	4.4406	4.5246	4.8670	4.5950	4.3263	4.4893	4.7316	4.6545	4.7989	4.3893	3.8881
recall@50	6.7364	6.7530	6.2164	6.0674	6.5484	6.0472	5.9646	6.3324	6.3860	**6.7565**	6.3307	6.2132	5.3603
recall@100	**10.9576**	9.8632	9.0345	9.3183	9.8335	8.6562	9.3201	9.0777	9.7607	10.1008	9.4100	9.2070	8.9121
ndcg@5	1.3717	1.3637	1.3777	1.3681	1.4067	1.4269	1.3568	1.3550	1.3868	1.3858	1.4413	**1.4573**	1.3632
ndcg@10	1.7071	1.7457	1.6559	1.7113	1.7259	1.7103	1.5983	1.6936	1.7798	1.7694	1.7715	**1.7969**	1.6123
ndcg@30	2.5056	2.5208	2.3637	2.3816	2.4691	2.4286	2.3061	2.4063	2.4917	2.4435	**2.5217**	2.4266	2.2194
ndcg@50	**2.9653**	2.9365	2.8083	2.7707	2.8877	2.7944	2.7251	2.8576	2.9019	2.9565	2.9050	2.8741	2.6014
ndcg@100	**3.8442**	3.6277	3.4273	3.4760	3.6044	3.3925	3.4455	3.4395	3.6208	3.6672	3.5826	3.5443	3.3657
Process_data_time													
seconds	284.6269	280.6662	282.1414	281.6628	282.0380	284.5839	283.7131	288.1532	**277.1801**	277.9502	279.2389	279.0803	278.9957
hrs	0:04:45	**0:04:33**	0:04:33	0:04:35	0:04:36	0:04:37	0:04:38	0:04:49	0:04:50	0:04:52	0:04:52	0:04:52	0:04:52

Y_train_dim	(4000, 8422)	**Label Cardinality**	2.9568	**Training Samples/Label**	0.4749
Y_test_dim	(1000, 8422)	**Label Density**	3.5107E-04		

ผลการทดลองข้อมูลชุดเรียนรู้ สำหรับข้อมูล 6000 ตัวอย่าง

ตาราง 4.88 : เปรียบเทียบผลข้อมูล Linear 1-6000 ชุดเรียนรู้ สำหรับสัดส่วนการลดมิติ 0.5 ระหว่างแบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับ แบบจำลองตั้งต้น

Random Method	Linear 1-6000	Training Result											
XMLC-PAO Ratio	org	XMLC-PAO-1 : 0.5						XMLC-PAO-2 : 0.5					
Adam Learning	0.003	0.0015	0.003	0.006	0.008	0.01	0.02	0.0015	0.003	0.006	0.008	0.01	0.02
mrr	0.0482	0.0482	0.0323	0.0468	0.0485	0.0478	0.0480	**0.0491**	0.0403	0.0392	0.0457	0.0420	0.0407
recall@5	**2.5185**	1.8731	1.2876	2.3635	1.5957	1.6763	1.4772	1.6976	1.7448	1.9095	2.1673	1.7950	1.7501
recall@10	**3.3690**	2.6822	1.7087	3.0496	3.2519	2.3831	2.2145	3.3635	2.3682	2.4084	2.7498	2.4610	2.4122
recall@30	5.4572	5.5129	2.8585	5.1249	5.4346	4.7931	4.6182	**6.4968**	3.7109	3.6966	4.7482	4.3603	4.2076
recall@50	7.2665	8.2311	3.6920	6.3797	7.0426	6.2002	7.1504	**8.3754**	4.7883	4.6458	6.0897	5.3880	5.3141
recall@100	10.2055	**12.5633**	8.3549	10.9953	11.1879	9.9990	12.2605	12.0943	9.0886	9.4047	10.3651	9.7755	10.0481
ndcg@5	**2.0633**	1.8169	1.1901	2.0082	1.6932	1.7640	1.6575	1.7778	1.5539	1.5393	1.8490	1.6222	1.5666
ndcg@10	**2.5122**	2.2416	1.4251	2.3683	2.3855	2.0945	2.0741	2.4864	1.8829	1.8265	2.1781	1.9911	1.9176
ndcg@30	3.3539	3.3110	1.9169	3.2304	3.2395	2.9605	3.0196	**3.6996**	2.4534	2.4316	2.9273	2.7383	2.6062
ndcg@50	3.9388	4.0807	2.2324	3.7057	3.7729	3.4527	3.7597	**4.2989**	2.8235	2.8001	3.3885	3.1324	2.9979
ndcg@100	4.7227	5.2042	3.4351	4.8724	4.8573	4.4816	4.9856	**5.2473**	3.9312	3.9912	4.4910	4.2196	4.2009
Process_data_time													
seconds	322.2234	320.0941	**306.1104**	307.4387	309.2927	310.5418	309.5992	350.0211	338.4714	341.0316	342.0850	344.7982	346.1831
hrs	0:05:22	0:05:20	**0:05:06**	0:05:07	0:05:09	0:05:11	0:05:10	0:05:50	0:05:38	0:05:41	0:05:42	0:05:45	0:05:46

Y_train_dim	(4800, 7560)	Label Cardinality	4.2481	Training Samples/Label	0.6349
Y_test_dim	(1200, 7560)	Label Density	5.6192E-04		

ตาราง 4.89 : เปรียบเทียบผลข้อมูล Linear 1-6000 ชุดเรียนรู้ สำหรับสัดส่วนการลดมิติ 0.8 ระหว่างแบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับ แบบจำลองตั้งต้น

Random Method	Linear 1-6000	Training Result											
XMLC-PAO Ratio	org	XMLC-PAO-1 : 0.8						XMLC-PAO-2 : 0.8					
Adam Learning	0.003	0.0015	0.003	0.006	0.008	0.01	0.02	0.0015	0.003	0.006	0.008	0.01	0.02
mrr	0.0482	**0.0507**	0.0453	0.0206	0.0427	0.0399	0.0482	0.0290	0.0485	0.0470	0.0378	0.0462	0.0443
recall@5	**2.5185**	2.3349	1.7802	0.6025	1.8264	1.8416	2.3346	1.0606	1.8451	1.7094	1.2937	2.4307	1.9698
recall@10	**3.3690**	3.1160	2.7173	0.7600	2.3343	2.5224	2.9684	1.4479	2.5201	2.5374	2.2462	2.9855	2.5703
recall@30	5.4572	4.9909	4.1220	1.1487	4.4979	3.7895	5.0858	2.4302	5.1690	**5.9833**	3.6936	5.2042	4.3038
recall@50	7.2665	6.9099	5.2942	1.5142	5.5616	4.8695	6.6759	3.1420	6.7745	**8.3056**	4.9616	6.8978	5.8288
recall@100	10.2055	11.6996	9.1592	6.9399	7.3556	8.9267	11.0610	8.3029	11.2693	**12.6620**	10.1057	11.1930	10.0433
ndcg@5	2.0633	**2.0909**	1.6665	0.5669	1.6360	1.5890	1.9832	0.9890	1.8397	1.8703	1.3152	2.0097	1.8068
ndcg@10	2.5122	**2.5355**	2.1233	0.6829	1.9483	1.9358	2.3531	1.1991	2.2117	2.3269	1.6968	2.3262	2.1493
ndcg@30	3.3539	3.3425	2.7817	0.9200	2.7812	2.5340	3.1625	1.6217	3.2022	**3.5378**	2.2801	3.2142	2.8960
ndcg@50	3.9388	3.9474	3.2176	1.0892	3.1537	2.9521	3.6704	1.9070	3.7776	**4.2286**	2.7153	3.7854	3.4322
ndcg@100	4.7227	5.1202	4.2220	2.4147	3.7340	4.0655	4.8406	3.2223	4.9582	**5.3088**	3.9715	4.9211	4.5315
Process_data_time													
seconds	322.2234	313.9615	314.3303	**313.3186**	314.4631	317.1575	314.7034	350.1158	349.9839	357.2913	353.7105	354.6048	354.0771
hrs	0:05:22	0:05:14	0:05:14	**0:05:13**	0:05:14	0:05:17	0:05:15	0:05:50	0:05:50	0:05:57	0:05:54	0:05:55	0:05:54

Y_train_dim	(4800, 7560)	Label Cardinality	4.2481	Training Samples/Label	0.6349
Y_test_dim	(1200, 7560)	Label Density	5.6192E-04		

ตาราง 4.90 : เปรียบเทียบผลข้อมูล Linear 1-6000 ชุดเรียนรู้ สำหรับสัดส่วนการลดมิติ 1.0 ระหว่างแบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับ แบบจำลองตั้งต้น

Random Method	Linear 1-6000	Training Result											
XMLC-PAO Ratio	org	XMLC-PAO-1 : 1.0						XMLC-PAO-2 : 1.0					
Adam Learning	0.003	0.0015	0.003	0.006	0.008	0.01	0.02	0.0015	0.003	0.006	0.008	0.01	0.02
mrr	0.0482	0.0516	0.0436	0.0483	0.0434	0.0321	0.0476	0.0467	0.0421	0.0418	0.0356	0.0286	0.0478
recall@5	2.5185	2.6161	2.1392	2.1930	1.3795	1.2774	2.4741	1.5173	1.4669	1.7803	1.3146	0.8727	1.8382
recall@10	3.3690	3.7005	2.7213	2.9571	1.9586	1.6722	3.1907	2.7827	1.9078	2.2309	1.8579	1.1806	2.7754
recall@30	5.4572	6.4475	4.7258	4.9329	4.4280	2.6039	5.3458	4.9817	3.7395	3.5817	2.7575	1.9647	4.8133
recall@50	7.2665	8.3069	5.7802	6.9488	5.8963	3.1932	7.4225	6.3662	4.7741	4.7193	3.5893	2.5293	6.1287
recall@100	10.2055	12.8460	9.6876	11.9451	11.0643	8.2879	10.6468	10.8970	8.7936	10.0831	8.5373	8.2395	8.9163
ndcg@5	2.0633	2.1609	1.8128	1.9465	1.4842	1.1401	1.9978	1.6304	1.5473	1.6464	1.2551	0.8457	1.7805
ndcg@10	2.5122	2.6814	2.1232	2.3688	1.8106	1.3540	2.4095	2.1725	1.7910	1.9139	1.5309	1.0319	2.2062
ndcg@30	3.3539	3.7289	2.8503	3.2403	2.7050	1.7578	3.3223	3.0070	2.4949	2.5178	1.9896	1.3915	2.9516
ndcg@50	3.9388	4.3181	3.2251	3.8785	3.1837	2.0097	3.9877	3.4954	2.8869	2.8964	2.3076	1.6261	3.3858
ndcg@100	4.7227	5.4792	4.2431	5.0684	4.4349	3.2487	4.8754	4.6849	3.9359	4.2099	3.5192	2.9908	4.2490
Process_data_time													
seconds	322.2234	319.6888	320.3034	320.6595	322.1877	322.5688	322.2798	358.3766	359.9477	359.9815	362.3375	365.7862	364.7638
hrs	0:05:22	0:05:20	0:05:20	0:05:21	0:05:22	0:05:23	0:05:22	0:05:58	0:06:00	0:06:00	0:06:02	0:06:06	0:06:05

Y_train_dim	(4800, 7560)	Label Cardinality	4.2481	Training Samples/Label	0.6349
Y_test_dim	(1200, 7560)	Label Density	5.6192E-04		

ตาราง 4.91 : เปรียบเทียบผลข้อมูล Linear 6001-12000 ชุดเรียนรู้ สำหรับสัดส่วนการลดมิติ 0.5 ระหว่างแบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับ แบบจำลองตั้งต้น

Random Method	Linear 6001-12000	Training Result											
XMLC-PAO Ratio	org	XMLC-PAO-1 : 0.5						XMLC-PAO-2 : 0.5					
Adam Learning	0.003	0.0015	0.003	0.006	0.008	0.01	0.02	0.0015	0.003	0.006	0.008	0.01	0.02
mrr	0.0350	**0.0363**	0.0334	0.0343	0.0327	0.0348	0.0324	0.0340	0.0327	0.0341	0.0345	0.0347	0.0356
recall@5	1.7462	2.2891	1.6955	**2.2979**	1.7564	2.1123	1.4789	1.7604	1.7476	1.7903	2.0759	1.8399	2.1451
recall@10	2.7316	3.2763	2.9941	3.1598	2.9265	3.2870	2.7875	2.7529	2.7961	2.9487	3.2065	**3.3382**	3.1648
recall@30	**6.6770**	6.6224	6.2747	5.8553	5.6273	6.1102	5.5724	5.6127	6.3560	6.0999	6.5052	6.4835	6.2712
recall@50	**9.4773**	8.8804	8.3330	7.8514	7.4711	8.1438	7.4307	7.9254	8.6482	8.3880	8.7789	8.8169	8.8220
recall@100	**14.3466**	12.6156	12.4702	11.5952	11.3816	11.8283	11.0851	11.4157	12.7329	12.4591	12.8350	12.3294	13.2176
ndcg@5	1.5400	**1.8335**	1.5260	1.8178	1.5543	1.7591	1.4028	1.5839	1.5196	1.5668	1.6746	1.6132	1.7357
ndcg@10	1.9627	**2.2553**	2.0585	2.2037	2.0372	2.2252	1.9309	1.9966	1.9617	2.0161	2.1516	2.1889	2.1680
ndcg@30	3.0963	**3.2613**	3.0320	3.0743	2.8345	3.0981	2.7657	2.8874	2.9863	2.9658	3.1573	3.1427	3.1125
ndcg@50	3.7545	**3.8469**	3.5619	3.5560	3.3009	3.5987	3.2384	3.4725	3.5397	3.5267	3.7157	3.7101	3.7169
ndcg@100	**4.7932**	4.6888	4.4639	4.3495	4.1456	4.3968	4.0153	4.2558	4.4121	4.3818	4.5889	4.4649	4.6487
Process_data_time													
seconds	330.6347	314.0862	**313.1634**	314.4717	315.2333	316.7370	316.9835	341.7964	328.9384	330.5347	330.3362	332.1372	332.7932
hrs	0:05:31	0:05:14	**0:05:13**	0:05:14	0:05:15	0:05:17	0:05:17	0:05:42	0:05:29	0:05:31	0:05:30	0:05:32	0:05:33

Y_train_dim	(4800, 7914)	Label Cardinality	3.0110	Training Samples/Label	0.6065
Y_test_dim	(1200, 7914)	Label Density	3.8047E-04		

ตาราง 4.92 : เปรียบเทียบผลข้อมูล Linear 6001-12000 ชุดเรียนรู้ สำหรับสัดส่วนการลดมิติ 0.8 ระหว่างแบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับ แบบจำลองตั้งต้น

Random Method	Linear 6001-12000	Training Result											
XMLC-PAO Ratio	org	XMLC-PAO-1 : 0.8						XMLC-PAO-2 : 0.8					
Adam Learning	0.003	0.0015	0.003	0.006	0.008	0.01	0.02	0.0015	0.003	0.006	0.008	0.01	0.02
mrr	0.0350	0.0343	0.0349	0.0330	0.0306	**0.0352**	0.0345	0.0320	0.0347	0.0338	0.0321	0.0333	0.0328
recall@5	1.7462	1.7743	**2.2212**	2.1445	1.7279	2.2116	1.9461	2.1313	2.1738	1.5909	1.7311	1.8482	1.5865
recall@10	2.7316	2.9182	**3.4460**	3.0968	2.9937	3.2689	3.2194	3.3115	3.4223	2.5790	3.3420	2.9070	2.6553
recall@30	**6.6770**	6.3608	6.1468	5.9569	6.0046	6.1983	6.0661	6.5250	6.2579	5.4425	6.0380	5.6709	6.1897
recall@50	**9.4773**	8.5854	8.4929	8.3576	8.0457	8.5662	8.5583	8.8642	8.5698	8.0211	8.4109	7.7668	8.9351
recall@100	**14.3466**	12.9840	12.9768	12.7236	12.3224	13.0045	12.5956	12.9102	12.7699	12.5777	12.5676	12.0810	13.4506
ndcg@5	1.5400	1.5663	1.7543	1.7141	1.3096	**1.8100**	1.6539	1.6703	1.7240	1.4817	1.3889	1.5889	1.4933
ndcg@10	1.9627	2.0300	**2.2496**	2.1087	1.8359	2.2462	2.1799	2.1593	2.2415	1.9013	2.0022	2.0629	1.9405
ndcg@30	3.0963	3.0518	3.1129	2.9834	2.7421	**3.1397**	3.0652	3.1140	3.0991	2.8014	2.8597	2.9104	2.9954
ndcg@50	**3.7545**	3.6060	3.6799	3.5618	3.2494	3.7141	3.6637	3.6873	3.6310	3.4097	3.4412	3.4132	3.6509
ndcg@100	**4.7932**	4.5125	4.6239	4.4740	4.1404	4.6590	4.5295	4.5242	4.5275	4.3752	4.3118	4.3116	4.6126
Process_data_time													
seconds	330.6347	320.4890	**320.2704**	322.1011	322.4132	324.0909	324.4111	342.2251	342.9915	344.7878	344.9005	346.0650	346.9204
hrs	0:05:31	0:05:20	**0:05:20**	0:05:22	0:05:22	0:05:24	0:05:24	0:05:42	0:05:43	0:05:45	0:05:45	0:05:46	0:05:47

Y_train_dim	(4800, 7914)	Label Cardinality	3.0110	Training Samples/Label	0.6065
Y_test_dim	(1200, 7914)	Label Density	3.8047E-04		

ตาราง 4.93 : เปรียบเทียบผลข้อมูล Linear 6001-12000 ชุดเรียนรู้ สำหรับสัดส่วนการลดมิติ 1.0 ระหว่างแบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับ แบบจำลองตั้งต้น

Random Method	Linear 6001-12000	Training Result											
XMLC-PAO Ratio	org	XMLC-PAO-1 : 1.0						XMLC-PAO-2 : 1.0					
Adam Learning	0.003	0.0015	0.003	0.006	0.008	0.01	0.02	0.0015	0.003	0.006	0.008	0.01	0.02
mrr	**0.0350**	0.0343	0.0336	0.0349	0.0300	0.0342	0.0347	0.0343	0.0337	0.0318	0.0339	0.0348	0.0332
recall@5	1.7462	2.0718	2.1852	1.9300	1.5187	**2.2476**	1.8842	1.8520	2.1777	1.5578	1.6782	2.1576	1.8877
recall@10	2.7316	3.2701	3.0930	3.1174	2.7870	**3.3925**	3.0274	3.1194	3.2995	2.6280	2.6573	3.2434	2.9483
recall@30	6.6770	**6.9470**	6.0376	6.8252	5.8171	6.4144	6.1010	6.3107	6.0556	5.5653	5.8438	6.3739	5.8307
recall@50	**9.4773**	9.0235	8.1934	8.8440	8.4619	8.2789	8.6245	8.5754	8.6346	7.9932	7.9772	8.6214	8.0035
recall@100	**14.3466**	12.8852	12.6345	13.2059	12.0852	12.4261	12.5297	12.7302	13.3230	12.6474	12.3307	12.8081	11.8729
ndcg@5	1.5400	1.7209	1.7355	1.6234	1.2260	**1.7773**	1.6476	1.5879	1.6647	1.4556	1.5284	1.7501	1.5511
ndcg@10	1.9627	2.2023	2.1354	2.0596	1.7310	**2.2561**	2.0920	2.0887	2.1460	1.9214	1.9589	2.2151	1.9936
ndcg@30	3.0963	**3.2593**	3.0575	3.1841	2.6599	3.2162	3.0546	3.0739	3.0139	2.8141	2.8921	3.1824	2.8933
ndcg@50	3.7545	**3.7664**	3.5956	3.6855	3.2995	3.6871	3.6420	3.6502	3.6473	3.3885	3.4086	3.7373	3.4284
ndcg@100	**4.7932**	4.5955	4.5387	4.5835	4.0947	4.5396	4.4682	4.5639	4.6495	4.3300	4.3055	4.6319	4.2394
Process_data_time													
seconds	330.6347	**327.6195**	328.0285	327.9177	328.6652	330.4154	330.7495	351.4285	352.9922	352.4577	353.9505	356.0543	355.4368
hrs	0:05:31	**0:05:28**	0:05:28	0:05:28	0:05:29	0:05:30	0:05:31	0:05:51	0:05:53	0:05:52	0:05:54	0:05:56	0:05:55

Y_train_dim	(4800, 7914)	Label Cardinality	3.0110	Training Samples/Label	0.6065
Y_test_dim	(1200, 7914)	Label Density	3.8047E-04		

ตาราง 4.94 : เปรียบเทียบผลข้อมูล Linear 12001-18000 ชุดเรียนรู้ สำหรับสัดส่วนการลดมิติ 0.5 ระหว่างแบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับ แบบจำลองตั้งต้น

Random Method	Linear 12001-18000	Training Result											
XMLC-PAO Ratio	org	XMLC-PAO-1 : 0.5						XMLC-PAO-2 : 0.5					
Adam Learning	0.003	0.0015	0.003	0.006	0.008	0.01	0.02	0.0015	0.003	0.006	0.008	0.01	0.02
mrr	0.0279	0.0269	0.0269	0.0253	0.0278	0.0271	0.0290	0.0261	**0.0297**	0.0274	0.0252	0.0253	0.0228
recall@5	1.7009	1.5514	1.7960	1.3606	1.7449	1.7204	1.6983	1.4891	**1.8795**	1.6754	1.6112	1.5486	1.5105
recall@10	2.9013	2.6589	2.6136	2.3069	2.6694	2.9717	**3.0226**	2.4046	2.7091	2.7547	2.6093	2.4241	2.3532
recall@30	5.8143	5.5838	5.2087	5.2430	5.2606	5.4689	5.7473	**5.9356**	5.5818	5.4227	5.1799	5.1822	5.4063
recall@50	**7.7443**	7.1465	6.9577	7.2462	7.4554	7.0092	7.5310	7.6914	7.3123	7.2691	6.9742	6.9980	7.2691
recall@100	**11.7129**	10.1618	10.5751	10.8180	10.8298	9.9111	10.8020	10.7235	10.6191	10.6254	10.9825	10.5502	10.4679
ndcg@5	1.2798	1.2405	1.3969	1.1218	1.3754	1.2515	1.4086	1.1884	**1.5451**	1.3322	1.2955	1.1904	1.0867
ndcg@10	1.7427	1.6799	1.7518	1.5022	1.7576	1.7403	**1.9130**	1.5544	1.8864	1.7510	1.6828	1.5481	1.4273
ndcg@30	2.6093	2.5471	2.5233	2.3602	2.5076	2.4834	**2.7488**	2.5732	2.7096	2.5373	2.4678	2.3876	2.3469
ndcg@50	3.0895	2.9230	2.9545	2.8435	3.0329	2.8612	**3.1937**	2.9847	3.1346	2.9894	2.8943	2.8111	2.8124
ndcg@100	**3.9443**	3.5421	3.6943	3.6029	3.7397	3.4813	3.8748	3.6357	3.8555	3.7062	3.7100	3.5229	3.4759
Process_data_time													
seconds	340.4285	334.3774	**321.5686**	322.6687	325.6223	325.6117	327.0249	333.3282	322.7012	322.9989	324.4087	325.3682	326.3478
hrs	0:05:40	0:05:34	**0:05:22**	0:05:23	0:05:26	0:05:26	0:05:27	0:05:33	0:05:23	0:05:23	0:05:24	0:05:25	0:05:26

Y_train_dim	(4800, 8109)	Label Cardinality	2.6469	Training Samples/Label	0.5919
Y_test_dim	(1200, 8109)	Label Density	3.2641E-04		

ตาราง 4.95 : เปรียบเทียบผลข้อมูล Linear 12001-18000 ชุดเรียนรู้ สำหรับสัดส่วนการลดมิติ 0.8 ระหว่างแบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับ แบบจำลองตั้งต้น

Random Method	Linear 12001-18000	Training Result											
XMLC-PAO Ratio	org	XMLC-PAO-1 : 0.8						XMLC-PAO-2 : 0.8					
Adam Learning	0.003	0.0015	0.003	0.006	0.008	0.01	0.02	0.0015	0.003	0.006	0.008	0.01	0.02
mrr	**0.0279**	0.0263	0.0263	0.0264	0.0221	0.0260	0.0257	0.0260	0.0244	0.0276	0.0262	0.0275	0.0250
recall@5	1.7009	1.3922	1.6912	1.5837	1.5701	**1.7865**	1.4411	1.4413	1.6284	1.6650	1.4233	1.6487	1.2922
recall@10	2.9013	2.3485	2.7935	2.3490	2.3508	2.7455	2.4681	2.7227	2.5712	**2.9461**	2.4461	2.6124	2.4552
recall@30	**5.8143**	5.4397	5.4721	4.6194	4.2326	5.0002	4.7952	5.0449	4.5239	5.1502	4.8267	5.6378	4.9037
recall@50	**7.7443**	7.3079	7.2667	6.2829	6.4242	6.8509	6.3727	7.2414	6.2173	6.7674	6.6075	7.4453	6.7974
recall@100	**11.7129**	10.4615	10.7783	9.7341	10.4129	9.7718	9.6295	10.8077	9.2535	9.8641	10.4792	10.8312	10.2773
ndcg@5	1.2798	1.1682	1.2707	1.2342	1.1925	**1.3495**	1.1704	1.1646	1.2556	1.3035	1.1750	1.3110	1.1074
ndcg@10	1.7427	1.5437	1.7077	1.5511	1.5207	1.7455	1.5793	1.6464	1.6209	**1.8035**	1.5749	1.7022	1.5707
ndcg@30	**2.6093**	2.4607	2.5111	2.2440	2.0985	2.4097	2.2722	2.3626	2.2394	2.4952	2.2898	2.5869	2.3040
ndcg@50	**3.0895**	2.9184	2.9345	2.6423	2.6238	2.8526	2.6417	2.8906	2.6582	2.9058	2.7052	3.0413	2.7572
ndcg@100	**3.9443**	3.5845	3.6629	3.3546	3.4385	3.4677	3.3219	3.6227	3.3026	3.5490	3.4915	3.7473	3.4726
Process_data_time													
seconds	340.4285	**330.3542**	330.9353	333.1282	332.1346	333.6525	333.7004	336.2830	337.6706	338.7178	338.6511	340.4044	340.6453
hrs	0:05:40	**0:05:30**	0:05:31	0:05:33	0:05:32	0:05:34	0:05:34	0:05:36	0:05:38	0:05:39	0:05:39	0:05:40	0:05:41

Y_train_dim	(4800, 8109)	Label Cardinality	2.6469	Training Samples/Label	0.5919
Y_test_dim	(1200, 8109)	Label Density	3.2641E-04		

ตาราง 4.96 : เปรียบเทียบผลข้อมูล Linear 12001-18000 ชุดเรียนรู้ สำหรับสัดส่วนการลดมิติ 1.0 ระหว่างแบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับ แบบจำลองตั้งต้น

Random Method	Linear 12001-18000	Training Result											
XMLC-PAO Ratio	org	XMLC-PAO-1 : 1.0						XMLC-PAO-2 : 1.0					
Adam Learning	0.003	0.0015	0.003	0.006	0.008	0.01	0.02	0.0015	0.003	0.006	0.008	0.01	0.02
mrr	0.0279	0.0252	0.0269	**0.0279**	0.0250	0.0273	0.0256	0.0269	0.0254	0.0265	0.0276	0.0260	0.0268
recall@5	1.7009	1.7075	1.5895	1.9083	1.6101	1.4841	1.5044	**1.9149**	1.3595	1.5046	1.7046	1.8045	1.5922
recall@10	**2.9013**	2.6241	2.5082	2.8864	2.6516	2.2804	2.2869	2.6815	2.2565	2.3049	2.5978	2.7002	2.3826
recall@30	**5.8143**	5.1373	5.0124	5.5109	5.1234	4.6285	4.6032	5.0278	4.8850	5.2516	4.9780	5.3454	5.0792
recall@50	**7.7443**	6.5689	6.7082	7.5057	7.1439	6.5958	6.6839	7.1465	6.6934	7.1306	6.9657	7.4854	6.9936
recall@100	**11.7129**	10.0082	10.6291	10.7745	10.1494	10.3803	10.0580	10.8757	10.3291	10.9316	10.7034	10.7425	10.4328
ndcg@5	1.2798	1.2297	1.2394	**1.4720**	1.2400	1.2510	1.2729	1.4257	1.1199	1.2334	1.3436	1.3739	1.2993
ndcg@10	1.7427	1.6058	1.6283	**1.8746**	1.6290	1.5769	1.5918	1.7621	1.4796	1.5556	1.6936	1.7329	1.6398
ndcg@30	2.6093	2.3638	2.3766	**2.6605**	2.3923	2.2826	2.3080	2.4775	2.2523	2.4328	2.4191	2.5306	2.4570
ndcg@50	3.0895	2.7194	2.7954	**3.1382**	2.8563	2.7631	2.8091	2.9836	2.6748	2.8726	2.8908	3.0410	2.9350
ndcg@100	**3.9443**	3.4260	3.5990	3.8184	3.4784	3.5389	3.5161	3.7470	3.4224	3.6505	3.6577	3.6950	3.6573
Process_data_time													
seconds	340.4285	**335.1420**	338.6070	337.9653	339.4068	341.1133	340.4696	344.3207	345.6679	345.6973	347.3022	349.6038	348.6850
hrs	0:05:40	**0:05:35**	0:05:39	0:05:38	0:05:39	0:05:41	0:05:40	0:05:44	0:05:46	0:05:46	0:05:47	0:05:50	0:05:49

Y_train_dim	(4800, 8109)	Label Cardinality	2.6469	Training Samples/Label	0.5919
Y_test_dim	(1200, 8109)	Label Density	3.2641E-04		

ตาราง 4.97 : เปรียบเทียบผลข้อมูล Linear 18001-24000 ชุดเรียนรู้ สำหรับสัดส่วนการลดมิติ 0.5 ระหว่างแบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับ แบบจำลองตั้งต้น

Random Method	Linear 18001-24000	Training Result											
XMLC-PAO Ratio	org	XMLC-PAO-1 : 0.5						XMLC-PAO-2 : 0.5					
Adam Learning	0.003	0.0015	0.003	0.006	0.008	0.01	0.02	0.0015	0.003	0.006	0.008	0.01	0.02
mrr	0.0252	0.0233	0.0236	0.0234	0.0221	0.0215	0.0229	0.0236	0.0238	0.0190	**0.0266**	0.0225	0.0229
recall@5	1.4347	1.3262	1.4158	1.3011	1.4916	1.3374	1.2103	1.3696	1.5082	1.2726	**1.6122**	1.1979	1.3886
recall@10	2.5103	2.1471	**2.6677**	2.1815	2.6674	2.3632	2.1157	2.5572	2.5502	2.4276	2.6496	2.2832	2.2780
recall@30	**5.9650**	4.7290	5.4305	4.7029	4.9101	5.0099	5.1994	4.9864	4.8869	4.9281	5.6826	4.9968	5.6255
recall@50	**8.2695**	7.0676	7.1264	6.7428	6.4477	6.4709	7.2786	6.6203	6.8667	6.5168	7.5592	6.2568	7.3105
recall@100	**11.9170**	10.3426	10.1092	9.9694	8.8008	9.6389	10.6949	9.7008	9.9060	9.8974	10.9545	9.6661	10.3062
ndcg@5	1.1130	1.0913	1.0881	1.0657	1.0932	1.0182	1.0060	1.0834	1.1607	0.8939	**1.2909**	0.9882	1.0941
ndcg@10	1.5407	1.4195	1.5686	1.4268	1.5341	1.4033	1.3862	1.5443	1.5694	1.3386	**1.6917**	1.3984	1.4286
ndcg@30	2.5108	2.1793	2.3668	2.1730	2.1785	2.1768	2.2524	2.2672	2.2433	2.0905	**2.5862**	2.1847	2.3572
ndcg@50	**3.0492**	2.7210	2.7705	2.6520	2.5612	2.5300	2.7330	2.6669	2.7046	2.4461	3.0254	2.4965	2.7526
ndcg@100	**3.8299**	3.3797	3.3808	3.3051	3.0529	3.1723	3.4042	3.3015	3.3408	3.1227	3.6738	3.1748	3.3685
Process_data_time													
seconds	330.1702	**311.5902**	313.0920	313.6789	314.8519	315.4998	316.0043	326.9791	329.2990	329.9521	332.0146	331.7797	332.5393
hrs	0:05:30	**0:05:12**	0:05:13	0:05:14	0:05:15	0:05:15	0:05:16	0:05:27	0:05:29	0:05:30	0:05:32	0:05:32	0:05:33

Y_train_dim	(4800, 8116)	Label Cardinality	2.4935	Training Samples/Label	0.5914
Y_test_dim	(1200, 8116)	Label Density	3.0724E-04		

ตาราง 4.98 : เปรียบเทียบผลข้อมูล Linear 18001-24000 ชุดเรียนรู้ สำหรับสัดส่วนการลดมิติ 0.8 ระหว่างแบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับ แบบจำลองตั้งต้น

Random Method	Linear 18001-24000	Training Result											
XMLC-PAO Ratio	org	XMLC-PAO-1 : 0.8						XMLC-PAO-2 : 0.8					
Adam Learning	0.003	0.0015	0.003	0.006	0.008	0.01	0.02	0.0015	0.003	0.006	0.008	0.01	0.02
mrr	**0.0252**	0.0245	0.0194	0.0223	0.0229	0.0234	0.0234	0.0217	0.0231	0.0230	0.0238	0.0246	0.0242
recall@5	1.4347	**1.7241**	1.3131	1.4102	1.4051	1.6878	1.4223	1.3602	1.6363	1.3214	1.5110	1.5280	1.4575
recall@10	2.5103	**2.9104**	2.1296	2.4820	2.8493	2.7110	2.3633	2.3013	2.4421	2.3262	2.5871	2.5023	2.3609
recall@30	**5.9650**	5.7924	4.1607	5.2789	5.6151	5.5099	4.7120	5.2019	5.0052	4.7631	5.4710	5.4474	5.0340
recall@50	**8.2695**	7.4161	5.8570	7.0667	7.3775	7.1128	6.1202	7.2473	6.6487	6.4646	7.1429	7.2697	7.0942
recall@100	**11.9170**	10.0361	8.2028	10.0254	10.4964	10.1408	9.1338	10.5247	9.6503	9.8365	9.9270	10.4586	10.4926
ndcg@5	1.1130	1.2506	1.0612	1.0575	1.0670	**1.2727**	1.1099	1.0322	1.2040	1.0336	1.1380	1.1722	1.1573
ndcg@10	1.5407	**1.7156**	1.3758	1.4486	1.6016	1.6837	1.4752	1.3924	1.5319	1.4135	1.5580	1.5552	1.4989
ndcg@30	2.5108	**2.5620**	1.9729	2.2382	2.4047	2.4857	2.1668	2.2038	2.3130	2.1499	2.3840	2.3934	2.2591
ndcg@50	**3.0492**	2.9412	2.3613	2.6532	2.8031	2.8630	2.4977	2.6755	2.7100	2.5420	2.7824	2.7947	2.7434
ndcg@100	**3.8299**	3.4682	2.8611	3.2491	3.4414	3.4720	3.1274	3.3566	3.3250	3.2231	3.3557	3.4339	3.4437
Process_data_time													
seconds	330.1702	**319.1146**	320.3805	321.6605	321.8609	323.3271	324.2843	343.1406	343.6759	345.9554	345.9595	347.5853	347.2350
hrs	0:05:30	**0:05:19**	0:05:20	0:05:22	0:05:22	0:05:23	0:05:24	0:05:43	0:05:44	0:05:46	0:05:46	0:05:48	0:05:47

Y_train_dim	(4800, 8116)	Label Cardinality	2.4935	Training Samples/Label	0.5914
Y_test_dim	(1200, 8116)	Label Density	3.0724E-04		

ตาราง 4.99 : เปรียบเทียบผลข้อมูล Linear 18001-24000 ชุดเรียนรู้ สำหรับสัดส่วนการลดมิติ 1.0 ระหว่างแบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับ แบบจำลองตั้งต้น

Random Method	Linear 18001-24000	Training Result											
XMLC-PAO Ratio	org	XMLC-PAO-1 : 1.0						XMLC-PAO-2 : 1.0					
Adam Learning	0.003	0.0015	0.003	0.006	0.008	0.01	0.02	0.0015	0.003	0.006	0.008	0.01	0.02
mrr	0.0252	0.0219	0.0222	0.0223	0.0218	0.0250	0.0234	0.0238	0.0206	**0.0257**	0.0235	0.0240	0.0222
recall@5	1.4347	1.4491	1.3582	1.6051	1.4306	1.5680	1.4027	1.6341	1.0858	1.5603	1.6447	**1.6557**	1.4238
recall@10	2.5103	2.1860	2.4794	2.7172	2.3632	2.6277	2.6447	2.5597	2.2514	2.7241	**2.8779**	2.7740	2.1723
recall@30	**5.9650**	4.7039	4.8327	5.6658	5.7623	5.0854	5.4880	4.5966	5.2136	5.2760	5.4270	5.7193	4.5451
recall@50	**8.2695**	6.3802	6.8472	7.3308	7.5779	7.0981	6.9690	6.9030	7.3515	7.5887	7.1471	7.5424	6.5076
recall@100	**11.9170**	9.5507	9.5136	10.4530	11.0816	10.2547	10.1795	10.0542	10.4983	11.0343	10.5018	10.4074	10.0636
ndcg@5	1.1130	1.1725	1.0586	1.2135	1.0391	1.2403	1.1281	1.2676	0.8758	**1.3052**	1.1749	1.2602	1.0626
ndcg@10	1.5407	1.4723	1.4815	1.6462	1.4132	1.6496	1.6032	1.6246	1.3152	**1.7465**	1.6599	1.6842	1.3498
ndcg@30	2.5108	2.1872	2.1816	2.5193	2.3732	2.3773	2.4234	2.2254	2.1699	2.4983	2.4070	**2.5332**	2.0527
ndcg@50	**3.0492**	2.5869	2.6369	2.9065	2.7809	2.8153	2.7867	2.7578	2.6545	3.0231	2.8183	2.9779	2.5069
ndcg@100	**3.8299**	3.2383	3.1924	3.5365	3.4916	3.4509	3.4485	3.3964	3.2988	3.7184	3.5002	3.5746	3.2271
Process_data_time													
seconds	330.1702	**326.1686**	327.9499	328.5812	328.7279	330.8992	330.8118	351.4135	354.5476	353.6833	354.2561	357.0741	355.8982
hrs	0:05:30	**0:05:26**	0:05:28	0:05:29	0:05:29	0:05:31	0:05:31	0:05:51	0:05:55	0:05:54	0:05:54	0:05:57	0:05:56

Y_train_dim	(4800, 8116)	Label Cardinality	2.4935	Training Samples/Label	0.5914
Y_test_dim	(1200, 8116)	Label Density	3.0724E-04		

ตาราง 4.100 : เปรียบเทียบผลข้อมูล Linear 24001-27932 (3933 ตัวอย่าง) ชุดเรียนรู้ สำหรับสัดส่วนการลดมิติ 0.5 ระหว่างแบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับแบบจำลองตั้งต้น

Random Method	Linear 24001-27932	Training Result											
XMLC-PAO Ratio	org	XMLC-PAO-1 : 0.5						XMLC-PAO-2 : 0.5					
Adam Learning	0.003	0.0015	0.003	0.006	0.008	0.01	0.02	0.0015	0.003	0.006	0.008	0.01	0.02
mrr	0.0246	0.0242	0.0239	0.0246	0.0251	0.0243	0.0248	0.0222	**0.0261**	0.0255	0.0250	0.0243	0.0244
recall@5	1.4641	1.5314	1.5441	1.3387	1.5295	1.3896	1.3370	1.4281	**1.5698**	1.4758	1.4878	1.5018	1.3849
recall@10	2.3535	2.4752	2.4726	2.4269	2.4104	**2.5178**	2.1103	2.0614	2.1820	2.2948	2.3436	2.2246	2.1923
recall@30	4.8178	4.8979	4.9429	4.9526	4.5811	5.0362	4.6902	4.6005	**5.1483**	4.7651	4.6162	4.8373	5.0787
recall@50	6.9468	6.7628	6.7786	6.9406	6.6128	6.7961	6.7329	6.2153	**7.0871**	7.0328	6.7523	6.6794	7.0710
recall@100	**11.4889**	10.4446	9.6813	10.3264	9.9032	10.1031	10.8037	10.0787	10.7837	10.6694	10.7888	10.1726	10.7853
ndcg@5	1.1768	1.1805	1.1995	1.1326	1.2528	1.1357	1.1613	1.1385	**1.3319**	1.2681	1.2388	1.1696	1.1562
ndcg@10	1.5221	1.5446	1.5578	1.5550	**1.6083**	1.5596	1.4553	1.3885	1.5959	1.5785	1.5938	1.4530	1.4706
ndcg@30	2.2654	2.2639	2.2761	2.3041	2.2262	2.2948	2.2114	2.1351	**2.4640**	2.2762	2.2578	2.2257	2.3089
ndcg@50	2.7673	2.7149	2.6880	2.7809	2.7116	2.7045	2.6814	2.5161	**2.9178**	2.7937	2.7434	2.6483	2.7660
ndcg@100	3.6617	3.4556	3.2841	3.4809	3.4015	3.3888	3.4991	3.2830	**3.6760**	3.5235	3.5550	3.3408	3.5400
Process_data_time													
seconds	225.7609	224.8722	**212.2504**	212.8361	212.5401	212.3979	213.6050	232.1737	219.4581	219.2147	220.9835	221.4392	222.1600
hrs	0:03:46	0:03:45	**0:03:32**	0:03:33	0:03:33	0:03:32	0:03:34	0:03:52	0:03:39	0:03:39	0:03:41	0:03:41	0:03:42

Y_train_dim	(3146, 5983)	Label Cardinality	2.4619	Training Samples/Label	0.5258
Y_test_dim	(787, 5983)	Label Density	4.1148E-04		

ตาราง 4.101 : เปรียบเทียบผลข้อมูล Linear 24001-27932 (3933 ตัวอย่าง) ชุดเรียนรู้ สำหรับสัดส่วนการลดมิติ 0.8 ระหว่างแบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับแบบจำลองตั้งต้น

Random Method	Linear 24001-27932	Training Result											
XMLC-PAO Ratio	org	XMLC-PAO-1 : 0.8						XMLC-PAO-2 : 0.8					
Adam Learning	0.003	0.0015	0.003	0.006	0.008	0.01	0.02	0.0015	0.003	0.006	0.008	0.01	0.02
mrr	0.0246	**0.0254**	0.0238	0.0239	0.0249	0.0249	0.0235	0.0247	0.0247	0.0247	0.0243	0.0242	0.0228
recall@5	1.4641	1.3871	1.3035	1.4540	1.5994	1.3875	1.6393	1.4632	1.5642	1.5022	1.3850	1.4681	**1.6664**
recall@10	2.3535	2.1682	2.1819	2.4420	**2.5270**	2.3243	2.3310	2.2865	2.3914	2.3262	2.3121	2.0611	2.3582
recall@30	4.8178	4.4403	4.5998	4.9965	**5.1343**	4.9043	4.4774	4.7178	4.9185	4.6314	4.9231	4.6804	4.9104
recall@50	6.9468	6.4365	6.2308	6.8930	**7.2828**	6.4546	6.3681	6.9380	7.0325	6.5312	7.1494	6.4258	6.8416
recall@100	**11.4889**	10.4105	9.6958	10.2295	10.5002	9.4848	9.7611	10.5682	10.7102	10.0933	9.9902	10.5397	10.5847
ndcg@5	1.1768	1.1687	1.0812	1.1620	1.1853	1.1572	1.2420	1.1654	1.1773	1.1862	1.1791	1.2116	**1.2571**
ndcg@10	1.5221	1.4860	1.4129	1.5503	**1.5562**	1.5379	1.5252	1.5013	1.5107	1.5312	1.5261	1.4680	1.5398
ndcg@30	2.2654	2.1758	2.1097	2.2933	**2.3269**	2.2739	2.1722	2.2502	2.2299	2.2232	2.2394	2.2325	2.2889
ndcg@50	2.7673	2.6505	2.5154	2.7446	**2.8322**	2.6569	2.6294	2.7707	2.7261	2.6626	2.7465	2.6436	2.7505
ndcg@100	**3.6617**	3.4691	3.2298	3.4416	3.5087	3.2910	3.3208	3.5176	3.5000	3.3820	3.3494	3.4406	3.5130
Process_data_time													
seconds	225.7609	**216.0009**	216.5009	218.6226	218.7450	219.5671	218.1574	224.9017	225.1814	225.7146	226.7655	226.9348	228.7295
hrs	0:03:46	**0:03:36**	0:03:37	0:03:39	0:03:39	0:03:40	0:03:38	0:03:45	0:03:45	0:03:46	0:03:47	0:03:47	0:03:49

Y_train_dim	(3146, 5983)	Label Cardinality	2.4619	Training Samples/Label	0.5258
Y_test_dim	(787, 5983)	Label Density	4.1148E-04		

ตาราง 4.102 : เปรียบเทียบผลข้อมูล Linear 24001-27932 (3933 ตัวอย่าง) ชุดเรียนรู้ สำหรับสัดส่วนการลดมิติ 1.0 ระหว่างแบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับแบบจำลองตั้งต้น

Random Method	Linear 24001-27932	Training Result											
XMLC-PAO Ratio	org	XMLC-PAO-1 : 1.0						XMLC-PAO-2 : 1.0					
Adam Learning	0.003	0.0015	0.003	0.006	0.008	0.01	0.02	0.0015	0.003	0.006	0.008	0.01	0.02
mrr	0.0246	0.0242	**0.0258**	0.0244	0.0247	0.0252	0.0242	0.0257	0.0251	0.0239	0.0249	0.0238	0.0245
recall@5	1.4641	1.3918	1.4308	1.4866	1.5299	1.4334	1.5028	1.5584	**1.6021**	1.4280	1.4923	1.2945	1.5160
recall@10	2.3535	2.1306	2.3448	2.4273	2.0676	2.4432	2.3253	2.2466	**2.6569**	2.4170	2.5227	1.9223	2.4146
recall@30	4.8178	4.7475	5.2729	4.9739	4.6562	5.1888	4.2831	4.7716	**5.4130**	4.7408	4.5478	4.6259	4.6248
recall@50	6.9468	6.7096	**7.6236**	6.8800	6.4295	6.9329	6.3787	6.8312	7.3875	6.3993	6.4321	6.2204	6.9812
recall@100	**11.4889**	10.8637	11.1946	10.7224	10.1443	10.7225	9.6642	10.5714	10.7004	9.8433	9.8466	10.3104	10.5894
ndcg@5	1.1768	1.1450	1.1736	1.1783	1.2425	1.1527	1.1674	**1.2763**	1.2367	1.1705	1.2049	1.1239	1.2296
ndcg@10	1.5221	1.4415	1.5281	1.5446	1.4806	1.5465	1.4884	1.5723	**1.6268**	1.5639	1.6006	1.3858	1.5729
ndcg@30	2.2654	2.1777	2.3421	2.2879	2.2579	2.3498	2.0637	2.2840	**2.4241**	2.2669	2.2138	2.1466	2.2197
ndcg@50	2.7673	2.6579	**2.8880**	2.7316	2.6876	2.7694	2.5696	2.7460	2.8561	2.6627	2.6453	2.5247	2.7505
ndcg@100	**3.6617**	3.5007	3.6025	3.5082	3.4189	3.5619	3.2328	3.5141	3.5211	3.3739	3.3589	3.3429	3.4789
Process_data_time													
seconds	225.7609	**219.6729**	221.3547	221.5515	222.9238	224.0557	225.0356	231.2841	233.6357	233.8867	233.5276	235.3800	236.5150
hrs	0:03:46	**0:03:40**	0:03:41	0:03:42	0:03:43	0:03:44	0:03:45	0:03:51	0:03:54	0:03:54	0:03:54	0:03:55	0:03:57

Y_train_dim	(3146, 5983)	Label Cardinality	2.4619	Training Samples/Label	0.5258
Y_test_dim	(787, 5983)	Label Density	4.1148E-04		

ผลการทดลองข้อมูลชุดเรียนรู้ สำหรับข้อมูล 7000 ตัวอย่าง

ตาราง 4.103 : เปรียบเทียบผลข้อมูล Linear 1-7000 ชุดเรียนรู้ สำหรับสัดส่วนการลดมิติ 0.5 ระหว่างแบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับ แบบจำลองตั้งต้น

Random Method	Linear 1-7000	Training result											
XMLC-PAO Ratio	org	XMLC-PAO-1 : 0.5						XMLC-PAO-2 : 0.5					
Adam Learning	0.003	0.0015	0.003	0.006	0.008	0.01	0.02	0.0015	0.003	0.006	0.008	0.01	0.02
mrr	0.0472	0.0434	0.0311	0.0483	0.0435	0.0369	0.0418	0.0447	**0.0483**	0.0430	0.0392	0.0442	0.0468
recall@5	**2.5190**	1.6568	1.2830	2.3196	2.1817	1.2383	1.7818	1.7445	2.2255	2.1789	1.3162	2.3240	2.2504
recall@10	**3.3894**	2.7000	1.6306	2.9780	2.8521	1.9938	2.3847	2.4066	3.1901	2.6748	1.9076	2.8353	2.9169
recall@30	5.0088	4.4140	2.4715	5.2324	5.0562	3.1263	4.0643	5.0499	5.5973	4.3704	3.2230	4.7296	**5.8096**
recall@50	7.3069	6.0378	3.0849	7.2743	6.3360	3.8956	5.3053	6.3509	**8.2360**	6.0513	4.5996	6.5905	8.1839
recall@100	11.6404	10.6537	7.7573	11.7092	10.6857	8.1334	9.2529	9.7817	12.0479	10.5361	9.0604	10.6686	**12.6146**
ndcg@5	**2.1017**	1.6937	1.1545	2.0958	1.8249	1.2794	1.7247	1.8419	1.9863	1.8999	1.4402	1.9407	1.9355
ndcg@10	**2.5370**	2.1368	1.3380	2.5132	2.1933	1.6142	2.1152	2.2171	2.4489	2.1964	1.7113	2.2563	2.3134
ndcg@30	3.2706	2.9009	1.7186	**3.4687**	3.0159	2.0924	2.8709	3.2005	3.3495	2.9477	2.2808	3.0678	3.3700
ndcg@50	3.9443	3.4278	1.9722	4.0827	3.4648	2.3773	3.3050	3.6754	**4.0908**	3.4960	2.7097	3.6738	4.0290
ndcg@100	5.0131	4.5131	3.1043	**5.2457**	4.5298	3.4238	4.3146	4.5553	5.1150	4.5821	3.8033	4.6904	5.1304
Process_data_time													
seconds	396.9514	384.4198	**372.3381**	374.0061	375.7066	376.0563	378.2971	391.6138	378.9385	380.4655	380.7068	386.5706	388.9956
hrs	0:06:37	0:06:24	**0:06:12**	0:06:14	0:06:16	0:06:16	0:06:18	0:06:32	0:06:19	0:06:20	0:06:21	0:06:27	0:06:29

Y_train_dim	(5600, 8465)	Label Cardinality	4.1248	Training Samples/Label	0.6615
Y_test_dim	(1400, 8465)	Label Density	4.8728E-04		

ตาราง 4.104 : เปรียบเทียบผลข้อมูล Linear 1-7000 ชุดเรียนรู้ สำหรับสัดส่วนการลดมิติ 0.8 ระหว่างแบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับ แบบจำลองตั้งต้น

Random Method	Linear 1-7000	Training result											
XMLC-PAO Ratio	org	XMLC-PAO-1 : 0.8						XMLC-PAO-2 : 0.8					
Adam Learning	0.003	0.0015	0.003	0.006	0.008	0.01	0.02	0.0015	0.003	0.006	0.008	0.01	0.02
mrr	**0.0472**	0.0341	0.0455	0.0447	0.0221	0.0447	0.0336	0.0366	0.0459	0.0440	0.0440	0.0435	0.0419
recall@5	**2.5190**	1.3578	1.9951	2.1855	0.6922	1.4279	1.2288	1.5179	2.2572	1.6409	1.6409	1.6051	1.6178
recall@10	**3.3894**	1.8932	2.9259	2.8187	0.9621	2.6311	1.9520	2.0468	3.1376	2.9115	2.9115	2.2500	2.3077
recall@30	5.0088	**5.2838**	4.8106	4.4633	1.5402	4.8177	2.9280	2.9331	5.0027	5.0273	5.0273	4.9747	4.2384
recall@50	7.3069	**7.3224**	6.0871	6.1006	1.8687	5.8550	3.7771	3.6976	6.7314	5.9561	5.9561	6.2966	5.5541
recall@100	**11.6404**	9.2248	8.7220	10.2762	6.9175	9.4834	8.3444	7.8515	10.0587	10.1518	10.1518	8.6902	9.9158
ndcg@5	**2.1017**	1.3323	1.8226	1.9167	0.6726	1.6335	1.2256	1.3870	1.9488	1.6613	1.6613	1.7927	1.6046
ndcg@10	**2.5370**	1.6545	2.3176	2.2944	0.8240	2.1729	1.5475	1.6647	2.3745	2.1685	2.1685	2.1441	1.9326
ndcg@30	**3.2706**	2.8430	3.1182	3.0219	1.0857	3.0556	1.9841	2.0802	3.1920	2.9306	2.9306	3.0875	2.6822
ndcg@50	**3.9443**	3.4508	3.5360	3.5323	1.2506	3.4567	2.2939	2.3664	3.7506	3.2891	3.2891	3.5052	3.1184
ndcg@100	**5.0131**	4.0135	4.2521	4.5895	2.4183	4.3826	3.3983	3.4287	4.6539	4.3066	4.3066	4.1400	4.2297
Process_data_time													
seconds	396.9514	380.5977	395.9614	383.9937	**379.9317**	382.0580	383.2983	400.6010	413.9514	404.6038	400.2305	403.3406	402.6027
hrs	0:06:37	0:06:21	0:06:36	0:06:24	**0:06:20**	0:06:22	0:06:23	0:06:41	0:06:54	0:06:45	0:06:40	0:06:43	0:06:43

Y_train_dim	(5600, 8465)	Label Cardinality	4.1248	Training Samples/Label	0.6615
Y_test_dim	(1400, 8465)	Label Density	4.8728E-04		

ตาราง 4.105 : เปรียบเทียบผลข้อมูล Linear 1-7000 ชุดเรียนรู้ สำหรับสัดส่วนการลดมิติ 1.0 ระหว่างแบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับ แบบจำลองตั้งต้น

Random Method	Linear 1-7000	Training result											
XMLC-PAO Ratio	org	XMLC-PAO-1 : 1.0						XMLC-PAO-2 : 1.0					
Adam Learning	0.003	0.0015	0.003	0.006	0.008	0.01	0.02	0.0015	0.003	0.006	0.008	0.01	0.02
mrr	0.0472	0.0409	0.0434	0.0432	0.0485	0.0370	0.0466	0.0493	0.0437	0.0250	0.0437	**0.0496**	0.0363
recall@5	2.5190	1.9121	2.1018	1.5481	2.1010	1.2319	1.7114	**2.5560**	1.6971	0.6550	1.7702	2.1259	1.1845
recall@10	3.3894	2.5870	3.0393	2.0580	2.9743	1.6825	2.3108	**3.5047**	2.7167	0.9655	2.1410	3.2122	1.8467
recall@30	5.0088	3.6420	5.1914	4.0846	5.8257	4.5714	5.0677	**6.5433**	4.6732	1.3726	3.6853	5.3165	2.8018
recall@50	7.3069	4.5237	6.8050	5.5755	7.6022	7.1167	6.7682	**8.6742**	6.4795	1.7335	4.6110	6.9078	3.5575
recall@100	11.6404	8.7432	10.5830	9.0106	11.2970	10.5828	10.9200	**13.4005**	10.4922	6.4312	10.3572	10.0967	8.0906
ndcg@5	2.1017	1.7187	1.7393	1.6474	1.9562	1.4091	1.7568	**2.1814**	1.6563	0.7737	1.8449	1.9652	1.2876
ndcg@10	2.5370	2.0766	2.1771	1.9466	2.4143	1.7057	2.0803	**2.6572**	2.1099	0.9228	2.1069	2.4814	1.5843
ndcg@30	3.2706	2.5996	2.9928	2.7368	3.4738	2.7613	3.0655	**3.7567**	2.8321	1.1657	2.8247	3.3686	2.0010
ndcg@50	3.9443	2.9187	3.5038	3.2033	4.0597	3.4734	3.6007	**4.3677**	3.3622	1.3180	3.2027	3.9082	2.2946
ndcg@100	5.0131	3.9473	4.4948	4.0908	5.0410	4.4237	4.6841	**5.5317**	4.4036	2.4817	4.5511	4.7825	3.3592
Process_data_time													
seconds	396.9514	**386.5607**	387.7175	391.6824	393.3755	395.8901	395.8560	403.0293	406.3500	408.3421	407.0961	409.3504	410.0739
hrs	0:06:37	0:06:21	0:06:36	0:06:24	**0:06:20**	0:06:22	0:06:23	0:06:41	0:06:54	0:06:45	0:06:40	0:06:43	0:06:43

Y_train_dim	(5600, 8465)	Label Cardinality	4.1248	Training Samples/Label	0.6615
Y_test_dim	(1400, 8465)	Label Density	4.8728E-04		

ผลการทดลองข้อมูลชุดเรียนรู้ สำหรับข้อมูลทั้งหมด

ตาราง 4.106 : เปรียบเทียบผลข้อมูล Linear 1-27932 ชุดเรียนรู้ สำหรับสัดส่วนการลดมิติ 0.5 ระหว่างแบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับ แบบจำลองตั้งต้น

Random Method	Linear 1-27932	Training Result											
XMLC-PAO Ratio	org	XMLC-PAO-1 : 0.5						XMLC-PAO-2 : 0.5					
Adam Learning	0.003	0.0015	0.003	0.006	0.008	0.01	0.02	0.0015	0.003	0.006	0.008	0.01	0.02
mrr	0.0285	0.0249	0.0224	0.0230	0.0237	0.0234	0.0237	0.0264	0.0261	0.0223	0.0260	0.0263	0.0251
recall@5	1.3014	1.3844	1.1508	0.9635	1.0167	1.3376	1.0508	1.4441	1.3462	1.1117	1.2512	1.4163	1.1336
recall@10	2.1553	1.7284	1.4854	1.2192	1.5302	1.7101	1.6168	2.0171	1.8757	1.4093	1.7532	2.1165	1.5163
recall@30	4.8234	2.6045	2.9430	2.4378	3.4364	3.4490	2.9137	3.3173	3.1777	3.0172	3.1866	3.6452	2.9092
recall@50	6.7657	3.6775	4.0196	3.8172	4.1664	4.3999	3.7440	4.7772	4.0425	3.7913	4.1060	4.7218	4.0878
recall@100	9.8766	5.6734	6.2580	5.6360	6.2023	5.9233	5.4282	7.0766	6.4498	5.4159	5.9496	7.2846	5.8562
ndcg@5	1.1668	1.2267	1.0265	1.0182	1.0275	1.0996	1.0472	1.2569	1.2209	1.0196	1.1787	1.2545	1.1381
ndcg@10	1.5155	1.3753	1.1812	1.1320	1.2298	1.2689	1.2783	1.5046	1.4484	1.1557	1.4145	1.5266	1.3186
ndcg@30	2.3045	1.6395	1.6329	1.5129	1.8044	1.8010	1.6688	1.8960	1.8319	1.6320	1.8637	1.9505	1.7651
ndcg@50	2.7618	1.8980	1.8953	1.8450	2.0012	2.0396	1.8736	2.2362	2.0461	1.8226	2.0970	2.2086	2.0373
ndcg@100	3.4402	2.3109	2.3682	2.2429	2.4320	2.3696	2.2384	2.7240	2.5270	2.1732	2.5001	2.7403	2.4082
Process_data_time													
seconds	1589.9088	1527.5139	1542.6340	1540.6797	1541.4248	1556.6501	1548.8470	2065.3880	2059.6768	2055.1546	2076.4978	2071.2446	2620.3325
hrs	0:26:30	0:18:33	0:18:30	0:18:24	0:18:09	0:18:06	0:18:01	0:21:10	0:20:53	0:20:52	0:20:39	0:20:42	0:20:27

Y_train_dim	(22345, 23335)	Label Cardinality	3.0060	Training Samples/Label	0.9576
Y_test_dim	(5587, 23335)	Label Density	1.2882E-04		

ตาราง 4.107: เปรียบเทียบผลข้อมูล Linear 1-27932 ชุดเรียนรู้ สำหรับสัดส่วนการลดมิติ 0.8 ระหว่างแบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับ แบบจำลองตั้งต้น

Random Method	Linear 1-27932	Training Result											
XMLC-PAO Ratio	org	XMLC-PAO-1 : 0.8						XMLC-PAO-2 : 0.8					
Adam Learning	0.003	0.0015	0.003	0.006	0.008	0.01	0.02	0.0015	0.003	0.006	0.008	0.01	0.02
mrr	0.0285	0.0265	0.0264	0.0267	0.0255	0.0256	0.0267	0.0262	0.0254	0.0225	0.0254	0.0263	0.0254
recall@5	1.3014	1.1822	1.2769	1.5860	1.1320	1.0430	1.3710	1.4652	1.3499	1.0555	1.2033	1.4734	1.1843
recall@10	2.1553	2.0191	1.9912	2.4487	1.6477	1.6246	2.0870	2.0002	1.8731	1.4757	1.7257	2.1704	1.6217
recall@30	4.8234	3.9544	3.3528	3.8917	3.3086	3.3161	3.2424	3.3720	3.1392	2.8911	3.2581	3.6387	3.0071
recall@50	6.7657	5.3895	4.3612	4.7105	4.6077	4.1332	4.7135	4.7515	4.0061	3.7433	4.1678	4.6741	3.8747
recall@100	9.8766	7.5734	6.1307	6.1203	6.3752	6.1691	6.8361	7.0706	6.2953	5.4006	5.8400	7.1634	5.6026
ndcg@5	1.1668	1.1439	1.1960	1.3058	1.1077	1.0585	1.2369	1.2635	1.1784	0.9954	1.1467	1.2721	1.1420
ndcg@10	1.5155	1.4707	1.4841	1.6589	1.3264	1.2838	1.5225	1.4926	1.4009	1.1874	1.3825	1.5304	1.3308
ndcg@30	2.3045	2.0392	1.8881	2.1324	1.8270	1.8239	1.8889	1.9003	1.7799	1.6276	1.8670	1.9443	1.7705
ndcg@50	2.7618	2.3664	2.1465	2.3478	2.1351	2.0333	2.2385	2.2271	1.9996	1.8503	2.0995	2.2044	1.9940
ndcg@100	3.4402	2.8364	2.5550	2.6616	2.5316	2.4830	2.6862	2.7192	2.4599	2.2095	2.4628	2.7262	2.3524
Process_data_time													
seconds	1589.9088	1608.5858	1611.3437	1585.3152	1607.1866	1609.5486	1610.5634	1976.7135	2021.4516	2001.7952	2004.8599	2011.1834	2488.9721
hrs	0:26:30	0:26:49	0:26:51	0:26:25	0:26:47	0:26:50	0:26:51	0:32:57	0:33:41	0:33:22	0:33:25	0:33:31	0:41:29

Y_train_dim	(22345, 23335)	Label Cardinality	3.0060	Training Samples/Label	0.9576
Y_test_dim	(5587, 23335)	Label Density	1.2882E-04		

ตาราง 4.108 : เปรียบเทียบผลข้อมูล Linear 1-27932 ชุดเรียนรู้ สำหรับสัดส่วนการลดมิติ 1.0 ระหว่างแบบจำลอง XMLC-PAO ทั้ง 2 แบบ กับ แบบจำลองตั้งต้น

Random Method	Linear 1-27932	Training Result											
XMLC-PAO Ratio	org	XMLC-PAO-1 : 1.0						XMLC-PAO-2 : 1.0					
Adam Learning	0.003	0.0015	0.003	0.006	0.008	0.01	0.02	0.0015	0.003	0.006	0.008	0.01	0.02
mrr	**0.0285**	0.0263	0.0260	0.0239	0.0262	0.0251	0.0256	0.0263	0.0255	0.0228	0.0258	0.0265	0.0256
recall@5	1.3014	1.3352	1.3859	1.0985	1.2366	1.2038	1.0997	**1.4550**	1.3529	1.0938	1.3187	1.4276	1.1807
recall@10	2.1553	1.9973	1.9666	1.8667	1.7417	1.8625	1.7577	1.9757	1.8787	1.5219	1.6717	**2.1783**	1.6679
recall@30	**4.8234**	3.0176	2.6819	3.3423	3.3346	3.7713	3.1993	3.3612	3.2194	3.0140	3.1701	3.7369	3.0345
recall@50	**6.7657**	3.8097	3.5765	3.9213	4.7075	4.6646	4.2053	4.8006	3.9093	3.7892	4.2667	4.6817	3.8698
recall@100	**9.8766**	5.4475	5.9931	5.2651	6.7650	6.9038	5.9996	7.1444	6.4741	5.1745	6.1468	7.1258	5.7198
ndcg@5	1.1668	1.2173	1.2449	1.0451	1.1693	1.0888	1.0932	**1.2620**	1.1968	1.0116	1.2113	1.2542	1.1418
ndcg@10	1.5155	1.4898	1.5075	1.3625	1.3797	1.3683	1.3605	1.4935	1.4190	1.2077	1.3795	**1.5420**	1.3584
ndcg@30	**2.3045**	1.8219	1.7529	1.8273	1.8589	1.9536	1.8019	1.9055	1.8160	1.6569	1.8376	1.9871	1.7843
ndcg@50	**2.7618**	2.0240	1.9738	2.0032	2.1788	2.2081	2.0463	2.2464	1.9909	1.8438	2.0956	2.2260	2.0012
ndcg@100	**3.4402**	2.3763	2.4775	2.3232	2.6301	2.6653	2.4400	2.7275	2.5074	2.1728	2.4891	2.7370	2.3853
Process_data_time													
seconds	**1589.9088**	1645.8691	1616.3506	1651.1584	1665.4783	1651.1828	1650.3774	2035.4821	1992.6092	1994.1963	2010.8026	2028.2997	2518.3773
hrs	**0:26:30**	0:27:26	0:26:56	0:27:31	0:27:45	0:27:31	0:27:30	0:33:55	0:33:13	0:33:14	0:33:31	0:33:48	0:41:58

Y_train_dim (22345, 23335) **Label Cardinality** 3.0060 **Training Samples/Label** 0.9576
Y_test_dim (5587, 23335) **Label Density** 1.2882E-04

บทที่ 5
บทสรุปและข้อเสนอแนะ

วิทยานิพนธ์เล่มนี้ได้กล่าวถึงงานวิจัยที่เกี่ยวข้องกับ การประยุกต์ใช้แบบจำลองการเรียนรู้เชิงลึก (Deep Learning Model) กับการจำแนกประเภทข้อมูลหลายคำตอบ (Extreme Multi-label Classification) ซึ่งมุ่งหมายที่จะศึกษาการลดจำนวนมิติของข้อมูลคำตอบ เพื่อลดระยะเวลาในการเรียนรู้ และเพิ่มประสิทธิภาพการทำงานของแบบจำลอง ข้อมูลที่เลือกมาใช้ในงานวิจัยนี้คือ ชุดข้อมูลกระทู้ในชุมชนออนไลน์ สแทคโอเวอร์โฟลว์ (Stackoverflow Dataset) ระหว่างปี ค.ศ. 2008-2010 จากเว็ปไซต์ *https://www.kaggle.com/stackoverflow/stacksample/data*

การวิจัยได้ทำการทดสอบกับชุดข้อมูลที่มีจำนวนมิติของข้อมูลคำตอบแตกต่างกัน จากการเลือกข้อมูลจากชุดข้อมูล สแทคโอเวอร์โฟลว์ (Stackoverflow Dataset) ในแบบต่าง ๆ และมีจำนวนข้อมูลในแต่ละชุดแตกต่างกัน แล้วนำไปทดสอบกับโครงสร้างแบบจำลอง XMLC-PAO แบบที่ 1 และ แบบจำลอง XMLC-PAO แบบที่ 2 (ภาพที่ 3.10) จากนั้นจึงนำผลที่ได้จากแบบจำลองดังกล่าว มาเปรียบเทียบกับ ผลจากแบบจำลองตั้งต้น (ภาพที่ 3.9)

ปริมาณที่นำมาเปรียบเทียบประสิทธิภาพของแบบจำลอง ได้แก่ ค่าเฉลี่ยส่วนกลับของลำดับ (Mean Reciprocal Rank – MRR), รีคอลล์ที่ M ลำดับแรก (Recall@M), ความน่าจะเป็นสะสมที่ลดทอนด้วยค่าปกติ ที่ M ลำดับแรก (NDCG@M) และเวลาที่แบบจำลองใช้ในการเรียนรู้ (Training Time) ซึ่งสำหรับแต่ละชุดข้อมูล จะมีปัจจัยที่เหมาะสมกับสัดส่วนการลดมิติที่แตกต่างกันออกไป

5.1 อภิปรายผลการศึกษา

แบบจำลอง XMLC-PAO ทั้งแบบที่ 1 และแบบที่ 2 ให้ประสิทธิภาพการทำงานที่ดีกว่าแบบจำลองตั้งต้น สำหรับบางชุดข้อมูลเท่านั้น โดยขึ้นอยู่คุณภาพของชุดคำตอบสำหรับแต่ละชุดข้อมูล เมื่อทำการเปรียบเทียบผลจากชุดข้อมูลที่มีจำนวนข้อมูล 5000 ตัว สำหรับชุดข้อมูลที่มีความหนาแน่นของชุดข้อมูลคำตอบ (Label Density) มากกว่า 0.0004 และ มีสัดส่วนสมาชิกของชุดข้อมูลคำตอบ (Label Cardinality) มากกว่า 2.75 แบบจำลอง XMLC-PAO ทำงานได้มีประสิทธิภาพดีกว่าแบบจำลองตั้งต้น

เมื่อเปรียบเทียบผลการศึกษาของทุกชุดข้อมูล (ภาพที่ 4.4) ชุดข้อมูลที่ทำให้แบบจำลอง XMLC-PAO ทั้งแบบที่ 1 และแบบที่ 2 ทำงานได้ดีกว่าแบบจำลองตั้งต้น ยังคงมีความหนาแน่นของชุดข้อมูลคำตอบ (Label Density) มากกว่า 0.0004 (ยกเว้นชุดข้อมูลที่เลือกแบบเชิงเส้นตัวอย่างที่ 18001-24000 (Linear 18001-24000) ที่แบบจำลองตั้งต้น ให้ประสิทธิภาพที่ดีกว่า)

นอกจากนี้ระยะเวลาในการเรียนรู้ของแบบจำลอง XMLC-PAO ที่มีการลดสัดส่วนมิติของข้อมูลลง ยังไม่แสดงความแตกต่างกับการเรียนรู้ของแบบจำลองสำหรับแบบจำลองตั้งต้น อย่างมีนัยสำคัญตามที่ได้ตั้งสมมติฐานไว้ แม้ว่าแบบจำลอง XMLC-PAO แบบที่ 1 ใช้เวลาในการเรียนรู้ที่น้อยกว่าแบบจำลองตั้งต้น แต่แบบจำลอง XMLC-PAO แบบที่ 2 ใช้เวลาในการเรียนรู้ที่มากกว่าแบบจำลองตั้งต้นเล็กน้อย และแบบจำลองที่มีสัดส่วนการลดมิติที่คงเหลือน้อยกว่า จะใช้เวลาในการเรียนรู้ที่น้อยกว่า

วิธีการเลือกชุดข้อมูลเป็นอีกปัจจัยที่มีผลต่อคุณภาพของชุดคำตอบสำหรับแต่ละชุดข้อมูล จากผลการศึกษาของชุดข้อมูลที่มีจำนวนข้อมูล 5000 ตัว การเลือกข้อมูลแบบเชิงเส้น (Linear Sampling) และการเลือกข้อมูลแบบสุ่ม (Random Sampling) ทำให้ความสัมพันธ์ระหว่างสัดส่วนสมาชิกของชุดข้อมูลคำตอบ (Label Cardinality) และ ความหนาแน่นของชุดข้อมูลคำตอบ (Label Density) กระจายตัวแตกต่างกัน

เมื่อเปรียบเทียบผลการทดลองของทุกชุดข้อมูล ชุดข้อมูลที่มีจำนวนข้อมูล 5000 ตัว และ 6000 ตัว โดยการเลือกข้อมูลแบบเชิงเส้น (Linear Sampling) มีการกระจายตัวของ สัดส่วนสมาชิกของชุดคำตอบ (Label Cardinality) และความหนาแน่นของชุดข้อมูลคำตอบ (Label Density) อยู่บนเส้นตรงคนละเส้น สำหรับชุดข้อมูลที่มีจำนวนข้อมูล 5000 ตัว และมีการเลือกแบบสุ่ม (Random Sampling) มีการกระจายตัวของสัดส่วนสมาชิกของชุดคำตอบ (Label Cardinality) และความหนาแน่นของชุดข้อมูลคำตอบ (Label Density) อยู่บนเส้นตรงเดียวกันกับ ข้อมูลที่มีจำนวนตัวอย่าง 7000 ตัว

สำหรับแบบจำลองที่ใช้ชุดข้อมูลทั้งหมด 27932 ตัวอย่าง มีความหนาแน่นของชุดข้อมูลคำตอบ (Label Density) ที่น้อยที่สุดเมื่อเทียบกับชุดข้อมูลอื่น ๆ ซึ่งทำให้กระบวนการลดมิติของข้อมูลคำตอบมีความแม่นยำน้อยลง เป็นผลให้แบบจำลอง XMLC-PAO ทั้งสองแบบ ไม่สามารถทำงานได้ดีกว่าแบบจำลองตั้งต้น

นอกจากนี้ ความหนาแน่นของชุดข้อมูลคำตอบ (Label Density) ยังมีผลกับการลดสัดส่วนมิติชุดคำตอบ สำหรับชุดข้อมูลที่มีจำนวนข้อมูลน้อยกว่า 5000 ตัวอย่าง (ชุดข้อมูล Linear 25001-27932 และ Linear 24001-27932) ความหนาแน่นของชุดข้อมูลคำตอบ (Label Density) อยู่ระหว่าง 0.0004-0.0005 แบบจำลอง XMLC-PAO แบบที่ 1 ที่ลดมิติข้อมูลคำตอบเหลือ 50% มี

ประสิทธิภาพการทำงานที่ดีกว่าแบบจำลองตั้งต้น และสำหรับชุดข้อมูลที่มีจำนวนข้อมูล 5000 ตัวอย่าง แบบจำลอง XMLC-PAO แบบที่ 2 ที่ลดมิติชุดคำตอบเหลือ 80% และความหนาแน่นของชุดข้อมูลคำตอบ (Label Density) มากกว่า 0.0005 มีประสิทธิภาพการทำงานที่ดีกว่าแบบจำลองตั้งต้น เนื่องจากความหนาแน่นของชุดข้อมูลคำตอบ (Label Density) ที่สูงกว่าชุดข้อมูลอื่น ๆ ความแม่นยำในการลดมิติชุดคำตอบจึงสูงกว่า แต่ปัจจัยที่ทำให้แบบจำลอง XMLC-PAO แบบที่ 1 และแบบที่ 2 มีประสิทธิภาพที่แตกต่างกัน ยังไม่ชัดเจน เป็นไปได้ว่าอาจจะขึ้นกับจำนวนข้อมูลที่ใช้ในแบบจำลอง

5.2 สรุปผลการศึกษา

จากการศึกษาปัจจัยที่มีผลต่อประสิทธิภาพการทำงานของแบบจำลอง XMLC-PAO แบบที่ 1 และ XMLC-PAO แบบที่ 2 ได้ข้อสรุปหลายประการดังนี้

5.2.1. การทำงานของแบบจำลอง XMLC-PAO ทั้งแบบที่ 1 และแบบที่ 2 ขึ้นอยู่กับคุณภาพของชุดคำตอบของแต่ละชุดข้อมูล จากผลการศึกษาเปรียบเทียบของชุดข้อมูล แบบจำลอง XMLC-PAO ทั้งสองแบบจะมีประสิทธิภาพการทำงานที่ดีกว่าแบบจำลองตั้งต้น เมื่อความหนาแน่นของชุดข้อมูลคำตอบ (Label Density) มากกว่า 0.0004

5.2.2. ระยะเวลาในการเรียนรู้ของแบบจำลอง XMLC-PAO ไม่แสดงความแตกต่างกับการเรียนรู้ของแบบจำลองสำหรับแบบจำลองตั้งต้น อย่างมีนัยสำคัญ แบบจำลอง XMLC-PAO แบบที่ 1 ใช้เวลาในการเรียนรู้ที่น้อยกว่าแบบจำลองตั้งต้น แต่แบบจำลอง XMLC-PAO แบบที่ 2 ใช้เวลาในการเรียนรู้ที่มากกว่าแบบจำลองตั้งต้นเล็กน้อย และแบบจำลองที่มีสัดส่วนการลดมิติที่คงเหลือน้อยกว่า จะใช้เวลาในการเรียนรู้ที่น้อยกว่า

5.2.3. วิธีการเลือกชุดข้อมูล มีผลกับคุณภาพของข้อมูล การเลือกข้อมูลแบบเชิงเส้น (Linear Sampling) และการเลือกข้อมูลแบบสุ่ม (Random Sampling) ทำให้ลักษณะการกระจายตัวของความหนาแน่นของข้อมูลคำตอบ (Label Density) และ สัดส่วนสมาชิกของชุดคำตอบ (Label Cardinality) แตกต่างกัน มีผลต่อการลดสัดส่วนข้อมูลในแบบจำลอง XMLC-PAO

5.2.4. สัดส่วนการลดมิติข้อมูลที่เหมาะสม จะแตกต่างกันไปตามลักษณะการกระจายตัวของ ความหนาแน่นของข้อมูลคำตอบ (Label Density) และ สัดส่วนสมาชิกของชุดคำตอบ (Label

Cardinality) จึงไม่สามารถสรุปได้ว่า แบบจำลอง XMLC-PAO แบบใดที่มีประสิทธิภาพการทำงานที่ดีที่สุด

5.3 ข้อเสนอแนะ

5.3.1 งานวิจัยนี้ได้ทำการศึกษาโดยใช้ข้อมูลกระทู้ในชุมชนออนไลน์ สแทคโอเวอร์โฟลว์ (Stackoverflow Dataset) ระหว่างปี ค.ศ. 2008-2010 เพียงชุดข้อมูลเดียว หากมีการศึกษาโดยใช้ชุดข้อมูลที่แตกต่างกัน ผลลัพธ์ที่ได้อาจแตกต่างกันออกไป และอาจพบความน่าสนใจในมุมมองอื่น ๆ

5.3.2 งานวิจัยนี้ สามารถประยุกต์ใช้งานได้กับข้อมูลการวิเคราะห์ข้อความประเภทอื่น ๆ ของระบบแนะนำ เช่น การแนะนำหนังสือ, การแนะนำบทความ, การแนะนำบทเพลง เป็นต้น ในการแก้ปัญหาการขาดข้อมูลตั้งต้น (Cold Start Problem)

บรรณานุกรม

บรรณานุกรม

K. Halder, L. Poddar, and M.Y. Kan. 2018. Cold Start Thread Rec- ommendation as Extreme Multi-label Classification,Extreme. *Multilabel Classification for Social Media,* 1911-1918.

F. Tai, and H. Lin. 2010. Multi-label Classification with Principle Label Space Transformation. *2nd International Workshop on Learinng from Multi-Label data.*

M. Chen, C. Yang, J. Chen, and P. Yi. 2013. A Method to Solve Cold- Start Problem in Recommendation System based on Social Net- work Sub-community and Ontology Decision Model. *Atlantis Press*, 159-166.

E. Pacharawongsakda and T.Theeramunkong. 2013. Multi-Label Classification Using Dependent and Independent Dual Space Reduction. *The Computer Journal*, 56 (9), 1113-1135.

R. Babbar and B. Schö̤lkopf. 2018. Adversarial Extreme Multi-label Classification. *stat.ML*.

J. Wicker, B. Pfahringer, and S.Kramer. 2012. Multi-Label Classification Using Boolean Matrix Decomposition. *SAC', 12*. 179-186.

B. Datta. 1995. Numerical Linear Algebra and Applications. *Brooks/Cole Publishing*.

T. Mikolov et al. 2013. Distributed Representations of Words and Phrasesand their Compositionality.

W. Ling et al. 2015. Two/Too Simple Adaptations of Word2Vec for Syntax Problems. *2015 Conference of the North American Chapter of the Association for Computational Linguistics – Human Language Technologies.*

F. Bernardini et al. 2009. Cardinality and Density Measures and Their Influence to Multi-label Learning Methods

StackSample: 10% of Stack Overflow Q&A. สืบค้นจาก *https://www.kaggle.com/stackoverflow/stacksample/data*

Discounted Cumulative Gain. สืบค้นจาก https://en.wikipedia.org/wiki/Discounted_cumulative_gain

ภาคผนวก

ภาคผนวก ก
โปรแกรมภาษาไพทอน
และจำนวนพารามิเตอร์สำหรับแบบจำลองของแต่ละชุดข้อมูล

1. **โปรแกรมภาษาไพทอน สำหรับดึงข้อมูลดิบจากไฟล์ข้อมูล ที่เป็นการเลือกข้อมูลแบบเชิงเส้น (Linear Sampling Python Script for Extracting Raw Data From File)**

สำหรับแต่ละชุดข้อมูล จะมีการเปลี่ยนค่าตัวแปร `nrows` เป็น `5000, 6000, 7000` และ `None` เพื่อให้ได้จำนวนตัวอย่าง ตามที่ต้องการ นอกจากนี้ยังมีการเปลี่ยนค่าตัวแปร `skiprows` ให้เป็นช่วงต่าง ๆ กัน ตามชื่อของแต่ละชุดข้อมูล

```
import os
from os.path import dirname, abspath
from keras.preprocessing.text import Tokenizer,one_hot
import csv
import numpy as np
from keras.preprocessing.sequence import pad_sequences
import pickle
from sklearn import preprocessing
from sklearn.model_selection import StratifiedShuffleSplit,
train_test_split
#from urlparse import urlparse
import re
import random
import pandas as pd

from numpy import array
from numpy import diag
from numpy import dot
from numpy import zeros
from scipy.linalg import svd

DATA_DIR = '/deepnn/'
Model_DIR = '/deepnn/'
max_post_len = 30
EMBEDDING_DIM = 100
MAX_NB_WORDS = 50000
forum = 'stackoverflow_users_post_all' #without the file extension
nrows = 5000

def str2numlist(numstr):
    return [int(k) for k in numstr[1:-1].split(',')]

def read_file(fname,nrows):
        skiprows = [i for i in range(1,5000)] ####
        f = pd.read_csv(fname, encoding='latin1', sep = '\t',
skiprows = skiprows, nrows = nrows)
        f = f[f.OwnerUserId.notnull()]
        posts = list(f.Body)
        user_list = list(f.users_list)
        user_list = list(f.users_list.apply(str2numlist))
        return posts, user_list

def get_embeddings_index():
        with open('/deepnn/glove.6B.{}d.txt'.format(EMBEDDING_DIM)
,encoding='latin1') as f:
```

```
                embeddings_index = {}
                for line in f:
                        values = line.split(' ')
                        word = values[0]
                        coefs = np.asarray(values[1:],
dtype='float32')
                        embeddings_index[word] = coefs
        return embeddings_index


def plst(y, select_ratio):
    # Singular-value decomposition
    u, s, vt = svd(y)

    # select vt rows by value of select_ratio
    M = round(select_ratio*y.shape[1])
    vt_select = vt[:M,:]
    p = vt_select.T
    h = y.dot(p)

    return h, p, s

def process_data():
    file_path = DATA_DIR + forum+'.csv'
    tokenizer = Tokenizer(num_words=MAX_NB_WORDS)

    posts_all, user_list_all = read_file(file_path, nrows)
    for i in range(0,len(posts_all)):
        t = posts_all[i]
        t = t.lower()
        t = ' '.join(re.sub("!"," _exmark_ ",t).split())
        t = ' '.join(re.sub("\?"," _qmark_ ",t).split())
        t = ' '.join(re.sub("can't","can not",t).split())
        t = ' '.join(re.sub("won't","will not",t).split())
        t = ' '.join(re.sub("ain't","is not",t).split())
        t = ' '.join(re.sub("n't "," not ",t).split())
        t = ' '.join(re.sub(r'http\S+', '_url_', t).split())
        posts_all[i] = t

    # create training and testing vars
    posts, posts_test, user_list, user_list_test =
train_test_split(posts_all, user_list_all, test_size=0.2,
random_state=42)

    tokenizer.fit_on_texts(posts+posts_test)

    sequences_of_posts = tokenizer.texts_to_sequences(posts)
    sequences_of_posts =
pad_sequences(sequences_of_posts,maxlen=max_post_len)
    posts = np.asarray(sequences_of_posts)

    sequences_of_test_posts =
tokenizer.texts_to_sequences(posts_test)
    sequences_of_test_posts =
pad_sequences(sequences_of_test_posts,maxlen=max_post_len)
    posts_test = np.asarray(sequences_of_test_posts)
```

```
    word_index = tokenizer.word_index

    count = 0
    tcount = 0

    embeddings_index = get_embeddings_index()
    embedding_matrix = np.zeros((len(word_index) + 1, EMBEDDING_DIM))
    for word, i in word_index.items():
        embedding_vector = embeddings_index.get(word)
        if embedding_vector is not None:
            embedding_matrix[i] = embedding_vector
            count += 1
        else:
            tcount += 1
            for j in range(0,EMBEDDING_DIM):
                embedding_matrix[i][j] = random.randint(-
20000,+20000)/10000.0
        del embeddings_index


    # print 'Found %s unique tokens.' % len(word_index)
    le = preprocessing.LabelEncoder()
    le.fit(np.hstack(user_list+user_list_test))

    le_y = [le.transform(u) for u in user_list]
    le_y_test = [le.transform(u) for u in user_list_test]
    enc = preprocessing.MultiLabelBinarizer()
    enc.fit(le_y+le_y_test)
    y = enc.transform(le_y)
    y_test = enc.transform(le_y_test)

    h100_train, p100, s100 = plst(y, 1.0)
    h100_test = y_test.dot(p100)

    h80_train, p80, s80 = plst(y, 0.8)
    h80_test = y_test.dot(p80)

    h50_train, p50, s50 = plst(y, 0.5)
    h50_test = y_test.dot(p50)

    return {'embedding_matrix':embedding_matrix,
            'len_word_index':len(word_index),
            'x_train':posts,
            'x_test':posts_test,
            'y_train_list':le_y,
            'y_test_list':le_y_test,
            'y_train':y,
            'y_test':y_test,
            'h100_train':h100_train,
            'h100_test':h100_test,
            'h80_train':h80_train,
            'h80_test':h80_test,
            'h50_train':h50_train,
            'h50_test':h50_test,
            'p100': p100,
```

```
            's100': s100,
            'p80': p80,
            's80': s80,
            'p50': p50,
            's50': s50}
```

2. **โปรแกรมภาษาไพทอน สำหรับดึงข้อมูลดิบจากไฟล์ข้อมูล ที่เป็นการเลือกข้อมูลแบบสุ่ม (Random Sampling Python Script for Extracting Raw Data From File)**

   สำหรับชุดข้อมูลที่เลือกข้อมูลแบบสุ่ม ต้องมีการเปลี่ยนค่าตัวแปร `random_state` ในชุดคำสั่ง `sample` เป็น `42` และ `101`

```
import os
from os.path import dirname, abspath
from keras.preprocessing.text import Tokenizer,one_hot
import csv
import numpy as np
from keras.preprocessing.sequence import pad_sequences
import pickle
from sklearn import preprocessing
from sklearn.model_selection import StratifiedShuffleSplit,
train_test_split
#from urlparse import urlparse
import re
import random
import pandas as pd

from numpy import array
from numpy import diag
from numpy import dot
from numpy import zeros
from scipy.linalg import svd

DATA_DIR = '/deepnn/'
Model_DIR = '/deepnn/'
max_post_len = 30
EMBEDDING_DIM = 100
MAX_NB_WORDS = 50000
forum =  'stackoverflow_users_post_all' #without the file extension
nrows = 5000

def str2numlist(numstr):
    return [int(k) for k in numstr[1:-1].split(',')]

def read_file(fname,nrows):
    f = pd.read_csv(fname, encoding='latin1', sep = '\t', nrows =
None)
    f = f.sample(n=nrows, random_state = 101)
#     f = f.sample(n=nrows, random_state = 42)
```

```
    f = f[f.OwnerUserId.notnull()]
    posts = list(f.Body)
    user_list = list(f.users_list)
    user_list = list(f.users_list.apply(str2numlist))
    return posts, user_list

def get_embeddings_index():
        with open(DATA_DIR +
'glove.6B.{}d.txt'.format(EMBEDDING_DIM),encoding='latin1') as f:
                embeddings_index = {}
                for line in f:
                        values = line.split(' ')
                        word = values[0]
                        coefs = np.asarray(values[1:],
dtype='float32')
                        embeddings_index[word] = coefs
        return embeddings_index

def plst(y, select_ratio):
    # Singular-value decomposition
    u, s, vt = svd(y)

    # select vt rows by value of select_ratio
    M = round(select_ratio*y.shape[1])
    vt_select = vt[:M,:]
    p = vt_select.T
    h = y.dot(p)

    return h, p, s

def process_data():

    file_path = DATA_DIR + forum+'.csv'
    tokenizer = Tokenizer(num_words=MAX_NB_WORDS)

    #############
    # Read Data #
    #############
    posts_all, user_list_all = read_file(file_path, nrows)


    for i in range(0,len(posts_all)):
        t = posts_all[i]
        t = t.lower()
        t = ' '.join(re.sub("!"," _exmark_ ",t).split())
        t = ' '.join(re.sub("\?"," _qmark_ ",t).split())
        t = ' '.join(re.sub("can't","can not",t).split())
        t = ' '.join(re.sub("won't","will not",t).split())
        t = ' '.join(re.sub("ain't","is not",t).split())
        t = ' '.join(re.sub("n't ","' not ",t).split())
        t = ' '.join(re.sub(r'http\S+', '_url_', t).split())
        posts_all[i] = t


# create training and testing vars
```

```
    posts, posts_test, user_list, user_list_test =
train_test_split(posts_all, user_list_all, test_size=0.2,
random_state=42)

    tokenizer.fit_on_texts(posts+posts_test)

    sequences_of_posts = tokenizer.texts_to_sequences(posts)
    sequences_of_posts =
pad_sequences(sequences_of_posts,maxlen=max_post_len)
    posts = np.asarray(sequences_of_posts)

    sequences_of_test_posts =
tokenizer.texts_to_sequences(posts_test)
    sequences_of_test_posts =
pad_sequences(sequences_of_test_posts,maxlen=max_post_len)
    posts_test = np.asarray(sequences_of_test_posts)


    word_index = tokenizer.word_index

    count = 0
    tcount = 0

    embeddings_index = get_embeddings_index()
    embedding_matrix = np.zeros((len(word_index) + 1, EMBEDDING_DIM))
    for word, i in word_index.items():
        embedding_vector = embeddings_index.get(word)
        if embedding_vector is not None:
            embedding_matrix[i] = embedding_vector
            count += 1
        else:
            tcount += 1
                    for j in range(0,EMBEDDING_DIM):
                embedding_matrix[i][j] = random.randint(-
20000,+20000)/10000.0
    del embeddings_index


    le = preprocessing.LabelEncoder()
    le.fit(np.hstack(user_list+user_list_test))

    le_y = [le.transform(u) for u in user_list]
    le_y_test = [le.transform(u) for u in user_list_test]
    enc = preprocessing.MultiLabelBinarizer()
    enc.fit(le_y+le_y_test)
    y = enc.transform(le_y)
    y_test = enc.transform(le_y_test)

    h100_train, p100, s100 = plst(y, 1.0)
    h100_test = y_test.dot(p100)

    h80_train, p80, s80 = plst(y, 0.8)
    h80_test = y_test.dot(p80)

    h50_train, p50, s50 = plst(y, 0.5)
    h50_test = y_test.dot(p50)
```

```
    return {'embedding_matrix':embedding_matrix,
            'len_word_index':len(word_index),
            'x_train':posts,
            'x_test':posts_test,
            'y_train_list':le_y,
            'y_test_list':le_y_test,
            'y_train':y,
            'y_test':y_test,
            'h100_train':h100_train,
            'h100_test':h100_test,
            'h80_train':h80_train,
            'h80_test':h80_test,
            'h50_train':h50_train,
            'h50_test':h50_test,
            'p100': p100,
            's100': s100,
            'p80': p80,
            's80': s80,
            'p50': p50,
            's50': s50
            }
```

3. **โปรแกรมภาษาไพทอน สำหรับการสร้างแบบจำลองการเรียนรู้เชิงลึกสำหรับแบบจำลองตั้งต้นและแบบจำลอง XMLC-PAO โครงสร้างที่ 1 (Python script for creating deep learning model of original model and XMLC-PAO-1)**

ส่วนต้นของโปรแกรมนี้ จะต้องดึงข้อมูลที่ได้จัดการไว้ในโปรแกรมข้อ 1 หรือ ข้อ 2 มาใช้ ซึ่งต้องเปลี่ยนชื่อไฟล์ให้สอดคล้องกับข้อมูลที่จะนำมาใช้ ในส่วนของคำสั่งด้านล่างนี้ที่ขีดเส้นใต้เอาไว้

```
from sample_5K_splidvalid_DeepNN_PLST_fromTr import process_data,
EMBEDDIN_DIM, max_post_len
```

จากนั้นจึงสั่งให้โปรแกรมนี้ทำงาน

```
# floyd run --data kantarakornj/datasets/sample_5k/1/:deepnn --gpu --
follow "python sample_5K_DeepNN_Model_vary_seed.py"
import time
start_all = time.time()

from sample_5K_splitvalid_DeepNN_PLST_fromTr import process_data,
EMBEDDING_DIM, max_post_len

import keras.backend as K
from keras import regularizers
```

```
from keras.models import Sequential, load_model, Model
from keras.layers import Input, LSTM, multiply, concatenate, add
from keras.layers.embeddings import Embedding
from keras.layers.core import Lambda,Permute, Flatten, Dense,
Dropout, Activation, Reshape, RepeatVector, Masking
from keras.callbacks import ModelCheckpoint, CSVLogger,
LearningRateScheduler, ReduceLROnPlateau,LambdaCallback
from keras.layers.recurrent import LSTM, GRU
from keras.layers.wrappers import Bidirectional, TimeDistributed
from keras.optimizers import SGD, RMSprop, Adam
from keras.utils import np_utils
from os.path import dirname, abspath
from keras.preprocessing.sequence import pad_sequences
from os import listdir
import numpy as np
import h5py, pickle
from random import randint, choice, shuffle, sample
from sys import argv
from keras.layers.advanced_activations import PReLU
from keras.utils import plot_model
import random
from keras.layers.merge import Multiply
from sklearn.utils import class_weight
from keras.layers.convolutional import  Conv1D,MaxPooling1D


DATA_DIR = '/deepnn/'
Model_DIR = ''
max_len = max_post_len
gru_dim = 128
bigru_dim = 2*gru_dim
out_size = max_len*bigru_dim
clusters = 100

def sum_att(x):
        return K.sum(x,axis=1)

def stacking(x):
        return K.stack(x,axis=1)


# init_model_org
def init_model_org(num_users,data=None):
        print('Compiling model...')

        embedding_matrix, len_word_index = data
        print('building the model...')
        inp = Input(shape = (max_len,),name="input")

        x = Sequential()
        x.add(Embedding(len_word_index + 1,
                                                        EMBEDDING_DIM,

        #mask_zero=True,

        input_length=max_len,
```

```
        weights=[embedding_matrix],
                                                ) )
        x.add(Bidirectional(GRU(gru_dim, return_sequences = True)))
        x.add(Bidirectional(GRU(gru_dim, return_sequences = True)))

        post = x(inp)

        y = Sequential()
        y.add(Flatten(input_shape=(max_len,bigru_dim)))
        y.add(Dense(max_len,activation='softmax',
input_shape=(out_size,)))
        y.add(RepeatVector(bigru_dim))
        y.add(Permute((2,1)))


        posts = []
        for i in range(0,clusters):
                post_att = y(post)
                post_i = Multiply()([post_att,post])
                post_i =
Lambda(sum_att,output_shape=(bigru_dim,))(post_i)
                posts.append(post_i)

        posts =
Lambda(stacking,output_shape=(clusters,bigru_dim,))(posts)

        out =
TimeDistributed(Dense(1,name="dense1",activation='tanh')) (posts)
        out = Dropout(0.3, name='drop4') (out)
        out = Flatten()(out)

        output = Dense(num_users, activation ='sigmoid') (out)
        model = Model(inputs=[inp], outputs=[output])
        print(y.summary())
        return model

# init_model
def init_model(num_users,data=None):
        print('Compiling model...')

        embedding_matrix, len_word_index = data
        print('building the model...')
        inp = Input(shape = (max_len,),name="input")

        x = Sequential()
        x.add(Embedding(len_word_index + 1,
                                                EMBEDDING_DIM,

        #mask_zero=True,

        input_length=max_len,

        weights=[embedding_matrix],
                                                ) )
        x.add(Bidirectional(GRU(gru_dim, return_sequences = True)))
```

```
        x.add(Bidirectional(GRU(gru_dim, return_sequences = True)))

        post = x(inp)

        y = Sequential()
        y.add(Flatten(input_shape=(max_len,bigru_dim)))
        y.add(Dense(max_len,activation='softmax',
input_shape=(out_size,)))
        y.add(RepeatVector(bigru_dim))
        y.add(Permute((2,1)))


        posts = []
        for i in range(0,clusters):
                post_att = y(post)
                post_i = Multiply()([post_att,post])
                post_i =
Lambda(sum_att,output_shape=(bigru_dim,))(post_i)
                posts.append(post_i)

        posts =
Lambda(stacking,output_shape=(clusters,bigru_dim,))(posts)

        out =
TimeDistributed(Dense(1,name="dense1",activation='tanh')) (posts)
        out = Dropout(0.3, name='drop4') (out)
        out = Flatten()(out)

        output = Dense(num_users) (out)
        model = Model(inputs=[inp], outputs=[output])
        print(y.summary())
        return model

# Define 'get_data()' and 'eval()' Function

def get_data():
        data = process_data()
        return data

# Revised Eval Function

def eval_1(y_pred,y_true_list):
    kvals = [5,10,30,50,100]
    recall_k = []
    ndcg_k = []
    rr = []
    print('******\n y_pred:', y_pred.shape, '\n******')
#     print('******\n h_pred:', h_pred.shape, '\n******')

    for i in range(0,len(y_true_list)):
        y_t = y_true_list[i]
        y_t_len = len(y_t)
        y_p_index = np.flip(np.argsort(y_pred[i]),0)
        for i in range(0,len(y_p_index)):
            if y_p_index[i] in y_t:
                rr.append(1/float(i+1))
```

```
                break
        idcg = np.sum([1.0/np.log2(x+2) for x in range(0,y_t_len)])
        r = []
        ndcg = []
        for k in kvals:
            correct = len(np.intersect1d(y_t,y_p_index[0:k]))
            r.append(correct/float(y_t_len))
            dcg = 0
            for i in range(0,k):
                if y_p_index[i] in y_t:
                    dcg = dcg + 1.0/np.log2(i+2)
            ndcg.append(dcg/idcg)
        ndcg_k.append(ndcg)
        recall_k.append(r)

    print(" mrr = "+str(np.mean(rr)))
    recall = np.mean(recall_k,axis=0)*100
    ndcg = np.mean(ndcg_k,axis=0)*100
    for i in range(0,len(kvals)):
        print(" recall@"+str(kvals[i])+"= "+str(recall[i]))
        print(" ndcg@"+str(kvals[i])+"= "+str(ndcg[i]))

def recall(y_true, y_pred):
        # Count positive samples.
        c1 = K.sum(K.round(K.clip(y_true * y_pred, 0, 1)))
        c2 = K.sum(K.round(K.clip(y_pred, 0, 1)))
        c3 = K.sum(K.round(K.clip(y_true, 0, 1)))
        return c1/c3

def train_val(x_train, x_test, y_train, y_test, y_train_list,
y_test_list, h_train, h_test, p, adamlr, rmslr, plst_ratio):

    embedding_matrix = data['embedding_matrix']
    len_word_index = data['len_word_index']
    num_users = h_train.shape[1]

    rmsprop = RMSprop(lr= rmslr)
    adam = Adam(lr=adamlr)
    class_weights = [1.,10.]

    print('\n*** model_dnn_',plst_ratio, '***\n')
    print('\n*** adam learning rate ', adamlr, '***\n')

    #set random state
#     seed(1)
#     set_random_seed(2)

    model_svd = init_model(num_users,data=[embedding_matrix,
len_word_index])
    model_svd.compile(optimizer='adam',
                      metrics= ['acc'],
                      loss='mean_squared_error')

    start_train = time.time()
    model_svd.fit([x_train],[h_train],
            validation_data=([x_test],[h_test])
```

```
            ,class_weight=[class_weights]
            ,batch_size=64,epochs=20,verbose=1)
    end_train = time.time()

    training_time = end_train-start_train

    # Training Data
    h_tr_pred = model_svd.predict(x_train)
    y_tr_pred = h_tr_pred.dot(p.T)

    print('\n*** model_dnn_',plst_ratio, '***\n')
    print('\n*** adam learning rate ', adamlr, '***\n')
    print('\n Training_time:', training_time)
    print('\n Process_data_time:', process_data_time)
    print('h_tr_pred min:',h_tr_pred.min())
    print('h_tr_pred max:',h_tr_pred.max())
    print('h_train min:',h_train.min())
    print('h_train max:',h_train.max())
    print('y_tr_pred min:',y_tr_pred.min())
    print('y_tr_pred max:',y_tr_pred.max())

    print('y_train min:',y_train.min())
    print('y_train max:',y_train.max())

    mse_train = (np.square(y_train - y_tr_pred)).mean()
    print('mse_train:', mse_train)

    # Testing Data
    h_ts_pred = model_svd.predict(x_test)
    y_ts_pred = h_ts_pred.dot(p.T)


    print('h_ts_pred min:',h_ts_pred.min())
    print('h_ts_pred max:',h_ts_pred.max())
    print('h_test min:',h_test.min())
    print('h_test max:',h_test.max())
    print('y_ts_pred min:',y_ts_pred.min())
    print('y_ts_pred max:',y_ts_pred.max())


    print('y_test min:',y_test.min())
    print('y_test max:',y_test.max())

    mse_test = (np.square(y_test - y_ts_pred)).mean()
    print('mse_test:', mse_test)

    # Ratio 0.5: Eval
    print('\nTrain_Eval')
    eval_1(y_tr_pred,y_train_list)

    print('\nTest_Eval')
    eval_1(y_ts_pred,y_test_list)

    # save model_dnn_0.5

model_svd.save("model_dnn_"+plst_ratio+"_adamlr"+str(adamlr)+".h5")
```

```
# Get Data at different ratio ###################
# - 100%
# - 80%
# - 50%

'''y_train and y_test is 100% of y features

   h_train_80 and h_test_80 is 80% of y features

   h_train_50 and h_test_50 is 50% of y features '''

################## Process Data #################

start_process = time.time()
data = process_data()
end_process = time.time()

process_data_time = end_process-start_process
################################################

########### Set Parameter for Training ##########

x_train = data['x_train']
x_test = data['x_test']

y_train_list = data['y_train_list']
y_test_list = data['y_test_list']
y_train = data['y_train']
y_test = data['y_test']

rmslr = 0.003
plst_ratio = ['0.5','0.8','1.0']
h_train = [data['h50_train'], data['h80_train'], data['h100_train']]
h_test = [data['h50_test'], data['h80_test'], data['h100_test']]
p = [data['p50'], data['p80'], data['p100']]
adamlr = [0.0015, 0.003, 0.006, 0.008, 0.01, 0.02]

################################################

################### Training ###################
for i in range(0, len(plst_ratio)):
    for adlr in adamlr:
        train_val(x_train, x_test, y_train, y_test, y_train_list,
y_test_list, h_train[i], h_test[i], p[i], adlr, rmslr, plst_ratio[i])
################################################

############ Training Original Model ############

num_users = np.array(y_test).shape[1]
embedding_matrix = data['embedding_matrix']
len_word_index = data['len_word_index']

rmsprop = RMSprop(lr= rmslr)
adam = Adam(lr=0.003)
class_weights = [1.,10.]
```

```
model = init_model_org(num_users,data=[embedding_matrix,
len_word_index])
model.compile(optimizer='adam',
                          metrics= ['acc'],
                          loss='binary_crossentropy')

start_train = time.time()
model.fit([x_train],[y_train],
                validation_data=([x_test],[y_test])
                ,class_weight=[class_weights]
                ,batch_size=64,epochs=20,verbose=1)
end_train = time.time()

# check training time
training_time = end_train-start_train

# Training Data
y_tr_pred = model.predict(x_train)
# y100_tr_pred = h100_tr_pred.dot(p100.T)


print('\n*** model_dnn_org ***\n')

print('y_tr_pred min:',y_tr_pred.min())
print('y_tr_pred max:',y_tr_pred.max())

print('y_train min:',y_train.min())
print('y_train max:',y_train.max())

mse_train = (np.square(y_train - y_tr_pred)).mean()
print('mse_train:', mse_train)

# Testing Data
y_ts_pred = model.predict(x_test)

print('y_ts_pred min:',y_ts_pred.min())
print('y_ts_pred max:',y_ts_pred.max())


print('y_test min:',y_test.min())
print('y_test max:',y_test.max())

mse_test = (np.square(y_test - y_ts_pred)).mean()
print('mse_test:', mse_test)

# Ratio 0.5: Eval
print('\nTrain_Eval')
eval_1(y_tr_pred,y_train_list)

print('\nTest_Eval')
eval_1(y_ts_pred,y_test_list)

end_all = time.time()

all_process_time = end_all-start_all
```

```
print('All process runtime:', all_process_time)
print('Process data runtime:', process_data_time)
print('Training runtime:', training_time)
################################################
```

4. **โปรแกรมภาษาไพทอน สำหรับการสร้างแบบจำลองการเรียนรู้เชิงลึกสำหรับแบบจำลอง XMLC-PAO โครงสร้างที่ 2 (Python script for creating deep learning model of original model and XMLC-PAO-2)**

ส่วนต้นของโปรแกรมนี้ เป็นเช่นเดียวกับตัวโปรแกรมในข้อ 3 ที่จะต้องดึงข้อมูลที่ได้จัดการไว้ในโปรแกรมข้อ 1 หรือ ข้อ 2 มาใช้ ซึ่งต้องเปลี่ยนชื่อไฟล์ให้สอดคล้องกับข้อมูลที่จะนำมาใช้ ในส่วนของคำสั่งด้านล่างนี้ที่ขีดเส้นใต้เอาไว้

```
from sample_5K_splidvalid_DeepNN_PLST_fromTr import process_data,
EMBEDDIN_DIM, max_post_len
```

จากนั้นจึงสั่งให้โปรแกรมนี้ทำงาน

```
# floyd run --data kantarakornj/datasets/sample_5k/1/:fit_operator --
gpu --follow "python DeepNN_5K_sample_vary_seed_fit_operator.py"
import time
start_all = time.time()


from sample_5K_splitvalid_DeepNN_PLST_fromTr import process_data,
EMBEDDING_DIM, max_post_len


# library for setting random state
# from numpy.random import seed
# seed(1)
# from tensorflow import set_random_seed
# set_random_seed(2)

import keras.backend as K
from keras import regularizers
from keras.models import Sequential, load_model, Model
from keras.layers import Input, LSTM, multiply, concatenate, add
from keras.layers.embeddings import Embedding
from keras.layers.core import Lambda,Permute, Flatten, Dense,
Dropout, Activation, Reshape, RepeatVector, Masking
from keras.callbacks import ModelCheckpoint, CSVLogger,
LearningRateScheduler, ReduceLROnPlateau,LambdaCallback
from keras.layers.recurrent import LSTM, GRU
from keras.layers.wrappers import Bidirectional, TimeDistributed
from keras.optimizers import SGD, RMSprop, Adam
from keras.utils import np_utils
```

```
from os.path import dirname, abspath
from keras.preprocessing.sequence import pad_sequences
from os import listdir
import numpy as np
import h5py, pickle
from random import randint, choice, shuffle, sample
from sys import argv
from keras.layers.advanced_activations import PReLU
from keras.utils import plot_model
import random
from keras.layers.merge import Multiply
from sklearn.utils import class_weight
from keras.layers.convolutional import  Conv1D,MaxPooling1D



DATA_DIR = '/fit_operator/'
Model_DIR = ''
max_len = max_post_len
gru_dim = 128
bigru_dim = 2*gru_dim
out_size = max_len*bigru_dim
clusters = 100

def sum_att(x):
        return K.sum(x,axis=1)

def stacking(x):
        return K.stack(x,axis=1)

# init_model

def init_model(num_users, num_output, data=None):
    print('Compiling model...')

    embedding_matrix, len_word_index = data
    print('building the model...')
    inp = Input(shape = (max_len,),name="input")

    x = Sequential()
    x.add(Embedding(len_word_index + 1,
                    EMBEDDING_DIM,
                    #mask_zero=True,
                    input_length=max_len,
                    weights=[embedding_matrix],
                   ) )
    x.add(Bidirectional(GRU(gru_dim, return_sequences = True)))
    x.add(Bidirectional(GRU(gru_dim, return_sequences = True)))

    post = x(inp)

    y = Sequential()
    y.add(Flatten(input_shape=(max_len,bigru_dim)))
    y.add(Dense(max_len,activation='softmax',
input_shape=(out_size,)))
    y.add(RepeatVector(bigru_dim))
```

```
    y.add(Permute((2,1)))


    posts = []
    for i in range(0,clusters):
        post_att = y(post)
        post_i = Multiply()([post_att,post])
        post_i = Lambda(sum_att,output_shape=(bigru_dim,))(post_i)
        posts.append(post_i)

    posts =
Lambda(stacking,output_shape=(clusters,bigru_dim,))(posts)
    out =
TimeDistributed(Dense(1,name="dense1",activation='tanh'))(posts)
    out = Dropout(0.3, name='drop4')(out)
    out = Flatten()(out)
    out = Dense(num_users)(out)

    output = Dense(num_output, trainable = False)(out)
    model = Model(inputs=[inp], outputs=[output])
    print(y.summary())

    return model

# Define 'get_data()' and 'eval()' Function

def get_data():
        data = process_data()
        return data

def eval(h_train,x_test,y_test_list,p,model):
    kvals = [5,10,30,50,100]
    recall_k = []
    ndcg_k = []
    rr = []
    y_true_list = y_test_list
    h_pred = model.predict([x_test])

    ### Transform back to y_pred
    y_pred = np.around(h_pred.dot(p.T))
    print('******\n y_pred:', y_pred.shape, '\n******')
    print('******\n h_pred:', h_pred.shape, '\n******')

    for i in range(0,len(y_true_list)):
        y_t = y_true_list[i]
        y_t_len = len(y_t)
        y_p_index = np.flip(np.argsort(y_pred[i]),0)
        for i in range(0,len(y_p_index)):
            if y_p_index[i] in y_t:
                rr.append(1/float(i+1))
                # print('rr: ', rr, '\n')
                # print('1/', i+1)
                break
        idcg = np.sum([1.0/np.log2(x+2) for x in range(0,y_t_len)])
        r = []
        ndcg = []
```

```
        for k in kvals:
            correct = len(np.intersect1d(y_t,y_p_index[0:k]))
            r.append(correct/float(y_t_len))
            dcg = 0
            for i in range(0,k):
                if y_p_index[i] in y_t:
                    dcg = dcg + 1.0/np.log2(i+2)
            ndcg.append(dcg/idcg)
        ndcg_k.append(ndcg)
        recall_k.append(r)

# Revised Eval Function

def eval_1(y_pred,y_true_list):
    kvals = [5,10,30,50,100]
    recall_k = []
    ndcg_k = []
    rr = []
    print('******\n y_pred:', y_pred.shape, '\n******')

    for i in range(0,len(y_true_list)):
        y_t = y_true_list[i]
        y_t_len = len(y_t)
        y_p_index = np.flip(np.argsort(y_pred[i]),0)
        for i in range(0,len(y_p_index)):
            if y_p_index[i] in y_t:
                rr.append(1/float(i+1))
                break
        idcg = np.sum([1.0/np.log2(x+2) for x in range(0,y_t_len)])
        r = []
        ndcg = []
        for k in kvals:
            correct = len(np.intersect1d(y_t,y_p_index[0:k]))
            r.append(correct/float(y_t_len))
            dcg = 0
            for i in range(0,k):
                if y_p_index[i] in y_t:
                    dcg = dcg + 1.0/np.log2(i+2)
            ndcg.append(dcg/idcg)
        ndcg_k.append(ndcg)
        recall_k.append(r)

    print(" mrr = "+str(np.mean(rr)))
    recall = np.mean(recall_k,axis=0)*100
    ndcg = np.mean(ndcg_k,axis=0)*100
    for i in range(0,len(kvals)):
        print(" recall@"+str(kvals[i])+"= "+str(recall[i]))
        print(" ndcg@"+str(kvals[i])+"= "+str(ndcg[i]))

def recall(y_true, y_pred):
        # Count positive samples.
        c1 = K.sum(K.round(K.clip(y_true * y_pred, 0, 1)))
        c2 = K.sum(K.round(K.clip(y_pred, 0, 1)))
        c3 = K.sum(K.round(K.clip(y_true, 0, 1)))
        return c1/c3
```

```
def train_val(x_train, x_test, y_train, y_test, y_train_list,
y_test_list, h_train, h_test, p, adamlr, rmslr, plst_ratio):

    embedding_matrix = data['embedding_matrix']
    len_word_index = data['len_word_index']
    num_users = h_train.shape[1]

    rmsprop = RMSprop(lr= rmslr)
    adam = Adam(lr=adamlr)
    class_weights = [1.,10.]

    print('\n*** model_dnn_',plst_ratio, '***\n')
    print('\n*** adam learning rate ', adamlr, '***\n')


    model_svd = init_model(num_users, p.shape[0],
data=[embedding_matrix, len_word_index])
    model_svd.compile(optimizer='adam',
                      metrics= ['acc'],
                      loss='mean_squared_error')


    # add convertor to be a last layer of model
    w1 = p.T
    b1 = np.zeros((w1.shape[1],))

    model_svd.layers[208].set_weights([w1, b1])

    start_train = time.time()
    model_svd.fit([x_train],[y_train],
            validation_data=([x_test],[y_test])
            ,class_weight=[class_weights]
            ,batch_size=64,epochs=20,verbose=1)
    end_train = time.time()

    training_time = end_train-start_train

    # Training Data
    h_tr_pred = model_svd.predict(x_train)

    print('\n*** model_dnn_',plst_ratio, '***\n')
    print('\n*** adam learning rate ', adamlr, '***\n')
    print('\n Training_time:', training_time)
    print('\n Process_data_time:', process_data_time)
    print('\ny_tr_pred min:',h_tr_pred.min())
    print('y_tr_pred max:',h_tr_pred.max())
    print('y_train min:',y_train.min())
    print('y_train max:',y_train.max())

    mse_train = (np.square(y_train - h_tr_pred)).mean()
    print('mse_train:', mse_train)

    # Testing Data
    h_ts_pred = model_svd.predict(x_test)
```

```
    print('y_ts_pred min:',h_ts_pred.min())
    print('y_ts_pred max:',h_ts_pred.max())

    print('y_test min:',y_test.min())
    print('y_test max:',y_test.max())

    mse_test = (np.square(y_test - h_ts_pred)).mean()
    print('mse_test:', mse_test)

    # Ratio 0.5: Eval
    print('\nTrain_Eval')
    eval_1(h_tr_pred,y_train_list)

    print('\nTest_Eval')
    eval_1(h_ts_pred,y_test_list)

    # save model_dnn_0.5

model_svd.save("model_dnn_"+plst_ratio+"_adamlr"+str(adamlr)+".h5")

# Get Data at different ratio
# - 100%
# - 80%
# - 50%

'''y_train and y_test is 100% of y features

   h_train_80 and h_test_80 is 80% of y features

   h_train_50 and h_test_50 is 50% of y features '''

################## Process Data #################

start_process = time.time()
data = process_data()
end_process = time.time()

process_data_time = end_process-start_process
################################################

########### Set Parameter for Training ##########

x_train = data['x_train']
x_test = data['x_test']

y_train_list = data['y_train_list']
y_test_list = data['y_test_list']
y_train = data['y_train']
y_test = data['y_test']

rmslr = 0.003
plst_ratio = ['0.5','0.8','1.0']
h_train = [data['h50_train'], data['h80_train'], data['h100_train']]
h_test = [data['h50_test'], data['h80_test'], data['h100_test']]
p = [data['p50'], data['p80'], data['p100']]
adamlr = [0.0015, 0.003, 0.006, 0.008, 0.01, 0.02]
```

```
################################################

################### Training ###################
for i in range(0, len(plst_ratio)):
    for adlr in adamlr:
        train_val(x_train, x_test, y_train, y_test, y_train_list,
y_test_list, h_train[i], h_test[i], p[i], adlr, rmslr, plst_ratio[i])
################################################
```

5. **จำนวนพารามิเตอร์สำหรับแบบจำลองของแต่ละชุดข้อมูล (Number of parameters for each dataset model)**

จากภาพที่ 3.9 และ 3.10 ในบทที่ 3 โครงสร้างแบบจำลองมี 3 แบบ คือ

- แบบจำลองตั้งต้น (Original Model)
- แบบจำลอง XMLC-PAO แบบที่ 1 (XMLC-PAO-1)
- แบบจำลอง XMLC-PAO แบบที่ 2 (XMLC-PAO-2)

จำนวนพารามิเตอร์สำหรับแบบจำลองของแต่ละชุดข้อมูล สำหรับแต่ละแบบจำลองได้แสดงไว้ในตารางภาคผนวกดังรายละเอียดด้านล่าง

- ตารางภาคผนวกที่ 1: แสดงจำนวนพารามิเตอร์สำหรับแบบจำลองตั้งต้น และแบบจำลองที่มีสัดส่วนมิติคงเหลือ 0.5
- ตารางภาคผนวกที่ 2: แสดงจำนวนพารามิเตอร์สำหรับแบบจำลองที่มีสัดส่วนมิติคงเหลือ 0.8
- ตารางภาคผนวกที่ 3: แสดงจำนวนพารามิเตอร์สำหรับแบบจำลองที่มีสัดส่วนมิติคงเหลือ 1.0

ตารางภาคผนวกที่ 1: แสดงจำนวนพารามิเตอร์สำหรับแบบจำลองตั้งต้น และแบบจำลองที่มีสัดส่วนมิติคงเหลือ 0.5

Random Method	Original Model		XMLC-PAO-1 0.5		XMLC-PAO-2 0.5	
	Trainable	Non-trainable	Trainable	Non-trainable	Trainable	Non-trainable
Linear 1-5000	5,250,290	-	4,914,465	-	4,914,465	22,127,877
Linear 201-5200	5,302,585	-	4,966,962	-	4,966,962	22,091,304
Linear 801-5800	5,447,321	-	5,109,880	-	5,109,880	22,331,244
Linear 5001-10000	6,060,276	-	5,709,907	-	5,709,907	24,064,453
Linear 10001-15000	6,194,018	-	5,836,579	-	5,836,579	25,066,739
Linear 12201-17200	6,352,427	-	5,994,483	-	5,994,483	25,126,960
Linear 15001-20000	6,579,267	-	6,214,253	-	6,214,253	26,129,220
Linear 20001-25000	6,772,770	-	6,422,805	-	6,422,805	24,029,777
Linear 25001-27932	4,988,031	-	4,746,035	-	4,746,035	11,486,424
Random seed-42	6,463,607	-	6,035,973	-	6,035,973	35,861,980
Random seed-101	6,550,861	-	6,125,550	-	6,125,550	35,473,464
Linear 1-6000	5,967,299	-	5,585,519	-	5,585,519	28,584,360
Linear 6001-12000	6,837,553	-	6,437,896	-	6,437,896	31,323,612
Linear 12001-18000	7,208,048	-	6,798,493	-	6,798,493	32,881,995
Linear 18001-24000	7,503,555	-	7,093,697	-	7,093,697	32,942,844
Linear 24001-27932	6,036,222	-	5,734,131	-	5,734,131	17,907,119
Linear 1-7000	6,699,304	-	6,271,771	-	6,271,771	35,832,345
Linear 1-27932	15,191,097	-	14,012,394	-	14,012,394	145,654,501

ตารางภาคผนวกที่ 2: แสดงจำนวนพารามิเตอร์สำหรับแบบจำลองที่มีสัดส่วนมิติคงเหลือ 0.8

Random Method	XMLC-PAO-1 0.8		XMLC-PAO-2 0.8	
	Trainable	Non-trainable	Trainable	Non-trainable
Linear 1-5000	5,115,960	-	5,115,960	35,396,622
Linear 201-5200	5,168,356	-	5,168,356	35,343,428
Linear 801-5800	5,312,385	-	5,312,385	35,728,654
Linear 5001-10000	5,920,189	-	5,920,189	38,507,287
Linear 10001-15000	6,051,002	-	6,051,002	40,095,456
Linear 12201-17200	6,209,209	-	6,209,209	40,196,048
Linear 15001-20000	6,433,221	-	6,433,221	41,799,524
Linear 20001-25000	6,632,784	-	6,632,784	38,439,326
Linear 25001-27932	4,891,273	-	4,891,273	18,377,320
Random seed-42	6,292,513	-	6,292,513	57,370,700
Random seed-101	6,380,777	-	6,380,777	56,755,858
Linear 1-6000	5,814,587	-	5,814,587	45,730,440
Linear 6001-12000	6,677,670	-	6,677,670	50,111,448
Linear 12001-18000	7,044,226	-	7,044,226	52,611,192
Linear 18001-24000	7,339,632	-	7,339,632	52,705,304
Linear 24001-27932	5,915,325	-	5,915,325	28,640,621
Linear 1-7000	6,528,311	-	6,528,311	57,333,445
Linear 1-27932	14,719,743	-	14,719,743	233,073,364

ตารางภาคผนวกที่ 3: แสดงจำนวนพารามิเตอร์สำหรับแบบจำลองที่มีสัดส่วนมิติคงเหลือ 1.0

Random Method	XMLC-PAO-1 1.0		XMLC-PAO-2 1.0	
	Trainable	Non-trainable	Trainable	Non-trainable
Linear 1-5000	5,250,290	-	5,250,290	44,242,452
Linear 201-5200	5,302,585	-	5,302,585	44,175,962
Linear 801-5800	5,447,321	-	5,447,321	44,655,806
Linear 5001-10000	6,060,276	-	6,060,276	48,128,906
Linear 10001-15000	6,194,018	-	6,194,018	50,119,320
Linear 12201-17200	6,352,427	-	6,352,427	50,246,832
Linear 15001-20000	6,579,267	-	6,579,267	52,251,212
Linear 20001-25000	6,772,770	-	6,772,770	48,045,692
Linear 25001-27932	4,988,031	-	4,988,031	22,968,056
Random seed-42	6,463,607	-	6,463,607	71,715,492
Random seed-101	6,550,861	-	6,550,861	70,938,506
Linear 1-6000	5,967,299	-	5,967,299	57,161,160
Linear 6001-12000	6,837,553	-	6,837,553	62,639,310
Linear 12001-18000	7,208,048	-	7,208,048	65,763,990
Linear 18001-24000	7,503,555	-	7,503,555	65,877,572
Linear 24001-27932	6,036,222	-	6,036,222	35,802,272
Linear 1-7000	6,699,304	-	6,699,304	71,664,690
Linear 1-27932	15,191,097	-	15,191,097	291,339,582

ภาคผนวก ข

ผลงานตีพิมพ์

Program Book

The 14th International Joint Symposium on
Artificial Intelligence and Natural Language Processing
(iSAI-NLP 2019)

Combining Extreme Multi-label Classification and Principal Label Space Transformation for Cold Start Thread Recommendation

Kantarakorn Jitharn
Big Data Engineering Program
College of Innovative Technology and Engineering
Dhurakij Pundit University
Bangkok, Thailand
605162020009@dpu.ac.th

Eakasit Pacharawongsakda
Big Data Engineering Program
College of Innovative Technology and Engineering
Dhurakij Pundit University
Bangkok, Thailand
eakasit.pac@dpu.ac.th

***Abstract*—The recommendation system has been widely used in various areas, e.g., entertainment, education, and travel. However, this technique faces two main challenges which are Cold Start and High-Dimensionality problems. The cold start happens when the system does not have enough profile of new users; therefore, the system cannot recommend products to them. The second issue comes from the fact that there are a lot of distinct products or users to be recommended. Recently, Extreme Multi-label Classification (XMLC) has been applied to the recommendation system and addressed the Cold Start issue. However, the previous method still has a high-dimensionality issue. In this paper, we proposed a new approach, namely XMLC-PAO, which integrated label space reduction with XMLC. In more details, we transformed the recommendation problem to XMLC and applied Singular Value Decomposition (SVD) to generate reducing operator of label space (products' or users' label space). For the feature space, Deep Learning technique has been used to extract features from texts. From the experiments with Stackoverflow online forums dataset, we have found that the XMLC-PAO showed better performance in terms of RECALL@M and NDCG@M when the dimensions were reduced to 50% and 80% of the original size.**

***Index Terms*—extreme multi-label classification, recommendation system, cold start problem, singular value decomposition - SVD**

I. INTRODUCTION

Recommendation system is a tool for recommending users of what items they are looking for. It has been used in a variety of area such as movies, music, books, and research articles. However, there is a classic problem in recommendation system called "Cold Start" which is about a system that could not recommend new items to any users or could not suggest any items to new users as there is no previous interaction.

Examples of traditional cold start solutions are 1) 'Statistical model-based approach' which is using the corresponding probability distribution statistics user for projecting and initializing rates and high probability items to recommend, 2) 'Average approach' in which the original rating matrix is filled using the average of all ratings of the items before collaborative filtering, and 3) 'Mode approach' that user's predicted result score is calculated from the most often user rating. However, those methods still have an area of improvement in recommendation system [1].

Multi-label classification is a classification of items which have multiple correlated labels. It was initially studied by Schapire and Singer in text categorization [2]. Later many techniques in multi-label classification have been proposed for various applications such as semantic scene classification, music emotion categorization, automated tag recommendation, bioinformatics research and sentiment analysis [3]. Extreme multi-label classification is a multi-label classification that has extremely large set of target label [4] such as online forums which have been created for asking opinions or questions in any community.

The eXtreme Multi-Label Classification Method (XMLC) has been used in predicting a set of users (labels) who will want to respond to any new items in online forums. Stackoverflow online forum dataset could be used as items of the recommendation system. It is in a form of 'thread' starting by posting questions or asking others for opinions on a certain topic. While community members would be enable users to ask questions, the key of system is to ensure that the members find questions relevant to their interest to get them answered. Selecting a subset of users from the set of all users in the community poses significant challenges due to scalability and sparsity [5].

One possible approach to deal with High-Dimensionality issue which leads to scalability and sparsity in extreme multi-label classification is to transform into subset of original label hypercube view which has important correlations between labels before learning. Principal Label Space Transformation (PLST) is a simple and efficient method which relies on only singular value decomposition as the key step [6].

This paper aims to apply combination of eXtreme Multi-Label Classification and Principal lAbel space transfOrmation (XMLC-PAO) with the Stackoverflow online forum dataset. The input thread is processed by using stacked bi-directional

Gated Recurrent Units (Bi-GRU) architecture for text encoding along with cluster sensitive attention (CSA) for exploiting correlation along the large label space [5]. Application of Principal Label Space Transformation in reducing extreme multi-label space can reduce high-dimensionality problem before creating prediction network; however, deep neural network structure needs to be transformed from a classification model to a regression model. The XMLC-PAO models of different reducing dimensional ratio would be evaluated by values of MRR, RECALL@M, and NDGC@M comparing with the original model architecture [5].

II. BACKGROUND

Recommendation system is a system that tries to match user's profile (content-based filtering) or user's social environment and past behavior (collaborative filtering) with some reference characteristics that are related to item characteristics. User-item can be associated with various kinds of interactions such as ratings, bookmarks, purchase frequency, number of 'likes', number of page visits etc.

A. Cold Start Recommendation Problem

Cold start problem is a problem that recommendation systems could not match user's characteristic to any reference for recommending any items. There are three cases of cold start which are:

1 **New community:** It refers to a new recommendation system that has already had a catalog of items but has no users.
2 **New user:** If a new user has just registered, there is no interaction provided to the system.
3 **New item:** If a new item is added, there is no content information in the system.

The common problem of those cases is a lack of user-item interaction which makes it challenges to provide reliable recommendations (Fig.1). However, our framework study would only focus on a new item cold start scenario and create recommendation model to predict potential users to that item.

B. Definition of multi-label classification

Extreme multi-label classification (XMLC) refers to a task of assigning items into their most relevant subset of labels from an extremely large collection of class labels. The fundamental difference between multi-label classification and traditional binary or multi-class classification is that in multi-class classification only one among the possible labels applies to an item, whereas in multi-label classification the labels can be correlated with each other or have a subsuming relationship, and multiple labels can apply for an item (e.g., 'politics' and 'White House' for news articles, 'electron- ics', 'Samsung' and 'smartphone' for products, 'Eiffel tower' and 'vacation 2017' for images) [5].

Let $X = \mathbb{R}^P$ and $Y = \{0,1\}^K$ be an P-dimensional feature space and K-dimensional binary label space, where P is the number of features and K is a number of possible labels, i.e. classes. Let $D = \{\langle x_1, y_1\rangle, \langle x_2, y_2\rangle, ..., \langle x_q, y_q\rangle\}$ is a set of q objects (e.g. documents, images, etc.) in a training dataset, where $x_i \in X$ is a feature vector that represents an i^{th} object and $y_i \in Y$ is a label vector with the length of $K, [y_{i1}, y_{i2}, ..., y_{iK}]$. Here, y_{ij} indicates whether the i^{th} object belongs (1) or not (0) to the j^{th} class.

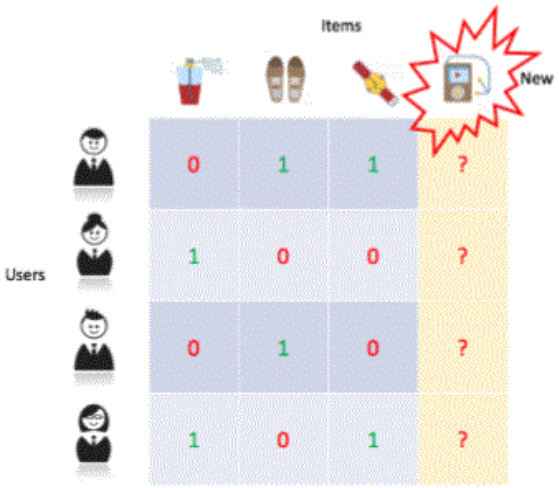


Fig. 1. Interaction Matrix of Cold Start Problem: '1' ⇒ interaction, '0' ⇒ no interaction

In general, 2 main phases are exploited in a multi-label classification problem: (1) model training phase and (2) classification phase. The goal of the model training phase is to build a classification model that can predict the label vector y_t for a new object with the feature vector x_t . This classification model is a mapping function $F : \mathbb{R}^P \rightarrow \{0,1\}^K$ which can predict a target value closest to its actual value in total. The classification phase uses this classification model to assign labels. For convenience, $X_{N\times P} = [x_1, ..., x_N]^T$ (i.e. D) denotes the feature matrix with N rows and P columns and $Y_{N\times K} = [y_1, ..., y_N]^T$ represents the *label matrix* with N rows and K columns, where $[-]^T$ denotes matrix transpose [3].

C. Input and Output of our system

Examples of input (thread) and output (user list) are showing in Table I. For the input part, there is a deep neural network architecture (Fig. 3) in extracting features which starts from word embedding, BiGRU and Cluster Sensitive Attention [5] before feeding into the predicting network. In the output part, user list is needed to reform to interaction matrix which has rows as items or threads and columns as user id. The interaction matrix would have value of 1 for interacting and 0 for not interacting between items and user ids corresponding for rows and columns. It is obviously sparse (Fig.2) which is a reason for trying to decrease dimension. In the next section, the interaction matrix would be called as label Y which would be referred in the experiment.

TABLE I
EXAMPLE OF RAW THREAD (INPUT) AND USER LIST (OUTPUT)

OwerID	Thread	User List
120	I am using CCNET on a...	[12734]
3400	How do you specify that ...	[419, 383, 60, ...]
580	Is it possible to do image...	[149, 34, 116, ...]
4320	Both the jQuery and Proto...	[17, 493, 598, ...]
5170	I have just started working ...	[225, 162, 667, ...]

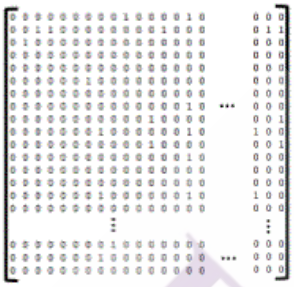


Fig. 2. Example of label Y: The column number is User ID and the row number is thread number

III. PROPOSED METHOD

A proposed method is a combination of Extreme Multi-Label Classification and Principal lAbel space transfOrmation (XMLC-PAO). Deep neural network in this paper would be created by using these following structures.

A. Principal Label Space Transformation

As mentioned in the previous section, the multi-label Y can be correlated with each other and the application of hypercube view can be applied to XMLC. The method of hypercube view used in this article is adjusted from original Principal Label Space Transformation (PLST) method. The Label Y hypercube with K-dimension would be decomposed into 3 parts through Singular Value Decomposition (SVD) for finding a proper projection matrix P and the decoder D for an M-flat label H [6].

This method is adjusted from original PLST method [6] to fit with experimental label Y dimension. In particular, the matrix Y (interaction matrix) is formed with each row being y_n, the occupied rows. Then, SVD would be performed on the $N \times K$ matrix Y to obtain three matrices [7].

$$Y = U\Sigma V^T \tag{1}$$

Here U is a N by N unitary matrix, Σ is a N by K diagonal matrix, and V^T is a K by K unitary matrix. Through SVD, each row y_n can be represented as a linear combination of singular vectors v_m^T in V^T. The vectors form a basis of a flat that passes through all the y_n. The matrix Σ is a diagonal matrix containing singular values σ_m that corresponds to the singular vectors v_m^T. It could be assumed that the singular values are ordered such that $\sigma_1 >= \sigma_2 >= ... >= \sigma_K$.

The equation (1) can be rewritten as

$$YV = U\Sigma \tag{2}$$

where the orthogonal basis V can be seen as a projection matrix of Y that maps each y to a different coordinate system. Since the largest M singular values correspond to the Principal directions of the original label space, the rest of the singular values could be discarded and their associated basis vectors in V obtained a smaller projection matrix

$$P = V_M = \left[\begin{array}{cccc} v_1 & v_2 & ... & v_M \end{array} \right] \tag{3}$$

that maps the vertices y to the M-flat label H which can be calculated as

$$H = Y.V_M \tag{4}$$

The transformed label H would be replaced label Y in the model structure which would change classifier model to regression model as value in H is not [0, 1].

An efficient decoder D for PLST can be calculated because $P = V$ is an orthogonal matrix, $P^{-1} = P^T$. This means V_M^T can be used to map any vector h on the M-flat space back to a point $h.V_M^T$ in label Y space.

TABLE II

Principal Label Space Transformation
1. With a parameter M, perform SVD on Y and obtain $V_M = \left[\begin{array}{cccc} v_1 & v_2 & ... & v_M \end{array} \right]$
2. Using $P = V_M$ for transform Y to H by $H = Y.V_M$
3. Using $D = V_M^T$ for transform H back to Y by $Y = H.V_M^T$

B. Feature Engineering

As each input of system is a post text, the feature engineering sections of deep neural network is needed. Main sections for extracting features are Text Encoding and Cluster Sensitive Attention.

1) Text Encoding: Each input post is consists sequence words $(w_1, w_2, ..., w_n)$. The first step of feature engineering is to embed each post into a lower-dimensional space which would be represented as a sequence of word vectors $\{q_1, q_2, ..., q_n\}$ where $q_i \in \mathbb{R}^d$. The word vectors use pre-trained GloVe embedding [8].

The post is then encoded using Bi-directional Gated Recurrent Unit (Bi-GRU) [9]. Input of this part is a sequence of word vector $\{q_1, q_2, ..., q_n\}$ and output is a sequence of $\{p_1, p_2, ..., p_n\}$ where $p_i \in \mathbb{R}^g$. A Bi-GRU reads the sequence of word vectors q_i from left to right in the forward stage to create p_i^f. The backward stage read p_i^f from forward stage

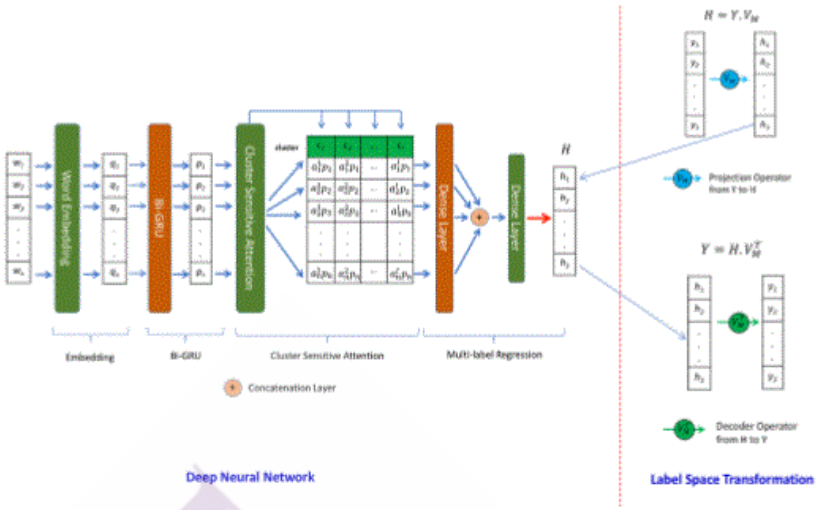


Fig. 3. Overall model architecture of XMLC-PAO

in reverse order to create p_i^b. The result from forward and backward states are concatenated to create the encoded hidden state of a post $p_i = [p_i^f; p_i^b]$ considering all its surrounding words.

2) Cluster Sensitive Attention: Cluster Sensitive Attention is a component in the network that can help focus on parts of post for different users [5]. To achieve this, it needs an *attention* mechanism that can give different weights to words of the posts and generate an encoded text representation using the weighted words, thus focusing on important parts. For each p_i of each input i from Bi-GRU component, a weight a_i for its corresponding word sequence w_i is computed for generating and attention vector $a = \{a_1, a_2, ..., a_n\}$ as:

$$a_i = \frac{exp(e_i)}{\sum_{j=1}^{n} exp(e_j)}, where \tag{5}$$

$$e_i = tanh(W_i p_i + b_i) \tag{6}$$

where W_i is a weight matrix of $1 \times g$ and b_i is the bias term. The text representation for a single attention layer (c_j) is then computed as:

$$c_j = \sum_{i}^{n} a_i^j p_i \tag{7}$$

The attention weights (a_i^j) should be dependent on different users' interests which can be softly clustered in a finite number of clusters and called it as "Cluster Sensitive Mechanism". From the text representation p_i, it can generate l different attention weight vectors $a_i^1, a_i^2, ..., a_i^l$. Thereafter, by using the different attention weights on p_i, a cluster sensitive encoding of the post text is $C = [c_1, c_2, ..., c_l]$.

C. Extreme Multi-Label Regression

For a post text, the l encoded texts are concatenated and fed through a fully connected layer with U output neurons (corresponding to each user). The fully connect layer learns the weights for its l inputs (corresponding to the different encoded texts).

$$z = tanh(W.C + b) \tag{8}$$

where W and b are weight and bias matrices respectively and $tanh$ is an element-wise non-linear activation function. The output of this feed-forward layer $z \in \mathbb{R}^U$ is then passed through output layer which having H label as output without any activation function to make it as regression model. Our model is trained using mean square error as the loss function. The network is end-to-end trainable and is optimized with Adam optimizer.

IV. EXPERIMENT

A. Dataset

Stackoverflow: is a dataset which has been used in this paper. It is a CQA website for programming related questions. It has been dumped from kaggle. All questions are posted during 2008 to 2010. The dataset would be used by removing all the code snippets (encapsulated within the tags $\langle code \rangle \langle /code \rangle$) from the question texts as showing in "Threads" column of Table I. The dataset has the whole average number of words in thread of about 94, average number of users per thread of 6.9, and sparisity ratio of 99.99%.

B. Evaluation

Before evaluation, the prediction from trained model need to be transformed from predicted H_{pred} to label Y space by applying decoder V_M^T as mentioned in proposed method section. The evaluation would be the comparison on Y and Y_{pred}.

$$Y_{pred} = H_{pred}.V_M^T \tag{9}$$

The label Y is huge and very high sparsity ratio (Fig.2). Therefore, it is a reason for not using overall accuracy as evaluation metric. It would be better to evaluate the positive instances i.e. the users who actually participated in a thread.

There are 3 following metrics considered as the better choices to evaluate the recommendation quality of the competing methods.

- Mean Reciprocal Rank (MRR) : indicates the position of the first relevant user in the ranked list.This measures the ability of a system in identifying an interested user at the top of the ranking. Let rt be the rank of the highest ranking relevant user for a test thread t. MRR is just the reciprocal rank, averaged over all threads in test set, n:

$$MRR = \frac{1}{n}\sum_{t=1}^{n}\frac{1}{r_t}$$

- Recall@M : considers how many top-M users actually interacted with the thread (the higher is better). Recall for the entire system is computed as the average recall value for all threads in test data.

- Normalized Discounted Cumulative Gain (NDCG@M) : is well suited for evaluation of recommendation system, as it rewards relevant results ranked higher in the returned list more heavily than those ranked lower. NDCG@M for a thread t is computed as:

$$NDCG_t = Z_t\sum_{j=1}^{M}\frac{2^{r(j)}-1}{log(1+j)}$$

  where Z_i is a normalized constant so that a perfect ordering would obtain NDCG of 1; each $r(j)$ is an integer relevance level (for used case, $r(j) = 1$ and $r(j) = 0$ for relevant and irrelevant recommendations, respectively) of result returned at the rank $j \in \{1, ..., k\}$. Then, for each M value, $NDCG_t$ is averaged over all (n) threads in the test set to get the overall NDCG@M.

The evaluation at $M = [5, 10, 30, 50, 100]$ are used to determine the quality of recommendation at different thresholds of the ranked list.

The Y_{pred} is a predicted interaction matrix of recommendation system (Fig. 2). The evaluation methods would rank columns of each row form each element value descending (the higher value is higher chance to recommend that user). The evaluation values would use ranked column index in the calculation as previously mentioned.

V. Results

The first 5,000 threads of data have been selected for training (4,000) and testing (1,000). By applying XMLC-PAO for reducing label Y dimension with ratio of [1.0, 0.8, 0.5] from original K-dimension (6,651 users), the results are verified with 1,000 testing data and compared with original model (not applying XMLC-PAO) [5]. All results are shown in Table III. As mentioned before, the accuracy is not used in evaluating result in the experiment. Predicting performance is based on value of MRR, RECALL@M, NDCG@M and training time. However, the training time and MRR for result of XMLC-PAO ratio 0.5 and 0.8 are obviously not better than ratio 1.0 and result without XMLC-PAO; the RECALL@M and NDCG@M of M = [30, 50, 100] for XMLC-PAO ratio 0.8 are much better than the original result.

All evaluation values have been adjusted to 0% (min) and 100% (max) which means they are quite small. It is a nature for this type of data due to the difficulty in processing. In accordance to this, there are some noises which are not removable. There are not so many users that are able to answer for any kind of questions. The higher RECALL@M means the more number of correct prediction results matching with true recommendation. The higher NDCG@M means the more correct prediction in higher rank matching with true result. The RECALL@M and NDCG@M of M = [30, 50, 100] for XMLC-PAO ratio 0.8 gain performance of 32.8% - 48.1% and 15.1% - 16.8% respectively. According to NDCG scores of this dataset, it can be concluded that XMLC-PAO ratio 0.8 model is able to correctly identify who are the interested users and places them near the top of the list. It is an efficient indicator for demonstrating better result from this experiment.

By adjusting Adam learning rate for deep neural network model of XMLC-PAO ratio 0.8 label (Table IV), all evaluation values are varied with different learning rates. The best result is still a model trained by Adam learning of 0.003. Almost all RECALL@M and NDGC@M are superior to others in all ways.

VI. CONCLUSIONS

Extreme multi-label (Y) of online Stackoverflow forums dataset has been transformed to subset of hypercube of original label (H) and used it for generating deep neural network. The experimental results of using different XMLC-PAO ratio label are compared with model using full original label and verified that the combination of Extreme multi-label and Principal Label Space Transform is successful in reducing the computational effort and achieves better performance.

The model still can be turned to give better prediction such as adjusting Adam learning rate. There are many points inside deep neural network architecture which can be changed and might give better results.

TABLE III
COMPARISON RESULTS OF TESTING DATA (1,000 THREADS)

Activation function	sigmoid	regression	regression	regression
Loss function	binary crossentropy	mse	mse	mse
XMLC-PAO Ratio	XMLC org	1.0	0.8	0.5
Adam Learing Rate	0.003	0.003	0.003	0.003
Training time	00:17:50	00:18:18	00:17:48	00:17:29
MRR	0.0302	0.0076	**0.0307**	0.0282
RECALL@M				
RECALL@5	1.1006	0.0704	1.0239	**1.1780**
RECALL@10	**1.6109**	0.0704	1.4403	1.4308
RECALL@30	1.9900	2.1058	**2.9463**	2.0056
RECALL@50	3.1718	3.4573	**4.2110**	2.8578
RECALL@100	4.4196	5.1964	**5.9378**	4.8524
NDCG@M				
NDCG@5	**1.1802**	0.0719	1.1274	1.1683
NDCG@10	**1.4150**	0.0719	1.3316	1.3077
NDCG@30	1.5474	0.7337	**1.8050**	1.4868
NDCG@50	1.8527	1.1032	**2.1325**	1.7113
NDCG@100	2.1727	1.5379	**2.5371**	2.1812

TABLE IV
COMPARISON RESULTS OF TESTING DATA (1,000 THREADS) WITH VARY ADAM LEARNING RATE IN COMPILING MODEL

Activation function	regression	regression	regression	regression	regression	regression
Loss function	mse	mse	mse	mse	mse	mse
XMLC-PAO Ratio	0.8	0.8	0.8	0.8	0.8	0.8
Adam Learing Rate	0.0015	0.003	0.006	0.008	0.01	0.02
Training time	00:18:02	00:17:48	00:17:54	00:17:47	00:17:44	00:17:38
MRR	0.0143	0.0307	0.0261	**0.0311**	0.0234	0.0216
RECALL@M						
RECALL@5	0.4167	1.0239	0.8571	**1.1409**	0.8663	0.9365
RECALL@10	0.7202	**1.4403**	1.0847	1.3648	1.2813	1.1351
RECALL@30	0.9234	**2.9463**	1.5775	2.5082	1.6429	1.6362
RECALL@50	1.8689	**4.2110**	2.1940	3.1395	2.6179	2.4456
RECALL@100	5.2004	**5.9378**	3.6647	4.5454	4.0976	3.8060
NDCG@M						
NDCG@5	0.4572	1.1274	0.9983	**1.2160**	0.8870	0.8735
NDCG@10	0.5833	**1.3316**	1.1004	1.3309	1.0661	0.9865
NDCG@30	0.6480	**1.8050**	1.2585	1.7478	1.2065	1.1587
NDCG@50	0.9052	**2.1325**	1.4101	1.9190	1.4932	1.4049
NDCG@100	1.6629	**2.5371**	1.7521	2.2448	1.8636	1.7273

This method can be applied to other kinds of data set such as product description, movie synopsis, or book content for solving in cold start recommendation.

ACKNOWLEDGMENT

Thanks to Kishaloy Halder for sharing his source code used in 'Cold Start Thread Recommendation as Extreme Multi-label Classification' research which can make this experiment started easier.

REFERENCES

[1] M. Chen, C. Yang, J. Chen, and P. Yi, "A Method to Solve Cold-Start Problem in Recommendation System based on Social Network Sub-community and Ontology Decision Model," Atlantis Press, pp. 159–166. 2013.

[2] R. Schapire and Y. Singer, "Boostexter: a boosting-based system for text categorization," Mach. Learn, Vol.39, pp. 135-168. 2000

[3] E. Pacharawongsakda and T.Theeramunkong, "Multi-Label Classification Using Dependent and Independent Dual Space Reduction," The Computer Journal, Vol. 56 No. 9, pp. 1113–1135. 2013

[4] R. Babbar and B. Schölkopf, "Adversarial Extreme Multi-label Classification," stat.ML, 2018

[5] K. Halder, L. Poddar, and M.Y. Kan, "Cold Start Thread Recommendation as Extreme Multi-label Classification," Extreme Multilabel Classification for Social Media, pp. 1911–1918. 2018.

[6] F. Tai and H. Lin, "Multi-label Classification with Principal Label Space Transformation," 2nd International Workshop on Learinng from Multi-Label data, Haifa, Israel, 2010.

[7] B. Datta, "Numerical Linear Algebra and Applications," Brooks/Cole Publishing, 1995.

[8] J. Pennington, R. Socher, and C. D Manning, "Glove: Global vectors for word representation," In Proc. of EMNLP, 2014.

[9] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of

ประวัติผู้เขียน

ชื่อ-นามสกุล	กันทรากร จิตต์หาญ
ประวัติการศึกษา	วิทยาศาสตรมหาบัณฑิต สาขาธรณีฟิสิกส์ มหาวิทยาลัยสงขลานครินทร์ ปีการศึกษา 2547
	วิทยศาสตรบัณฑิต สาขาฟิสิกส์ (เกียรตินิยมอันดับ 1) มหาวิทยาลัยมหิดล ปีการศึกษา 2545

ผลงานทางวิชาการ

- Kantarakorn Jitharn and Eakasit Pacharawongsakda (2019), Combining Extreme Multi-label Classification and Principal Label Space Transformation for Cold Start Thread Recommendation, 2019 14th International Joint Symposium on Artificial Intelligence and Natural Language Processing (iSAI-NLP)