Data Mining – 1

Why Mine Data?

- เอาข้อมูลมาวิเคราะห์ ทำนายอนาคต
- Mine อะไรจาก Data? – เพราะว่า Data เยอะ เก็บอยู่ใน warehouse มีข้อมูลหลายแบบ เยอะแยะ
- เมื่อก่อนคอมพ์แพง เดี๋ยวนี้คอมพ์ถูกเมมโมรี่ถูกลง Data ก็เก็บพอ แต่เราใช้ Algorithm มาช่วยด้วย เพื่อให้เก็บข้อมูลขนาดใหญ่ได้ดีขึ้น ประหยัด ใช้ Data Structure ที่เหมาะสม
- เกิดการแข่งขันกัน ทำให้เราต้องพยายามเอาข้อมูลมาวิเคราะห์ เพื่อให้เก่งกว่าคู่แข่ง

Mining Large Data Set – Motivation

- มักจะมีอะไรใหม่ๆ อยู่ใน data ที่เรามองไม่เห็น (hidden) ทำให้เราต้องดึงมันออกมา
- Uncover hidden knowledge

What is Data Mining?

- ขุดจาก Data ขนาดใหญ่ แล้วเอาไปตัดสินใจ
- ค้นข้อมูลเพื่อไปหาความสัมพันธ์ รูปแบบของข้อมูล
- เราไม่จำเป็นต้องวิเคราะห์ด้วย stat เลยในครั้งแรก เราทำ explore ดูก่อนก็ได้ ว่ามันมีอะไร แล้วค่อยเอาไปตัดสินใจ เช่น เราอยากดูการใช้โทรศัพท์ของ**คนไทย** แต่ข้อมูลที่เรามี มีแต่ข้อมูลของคน**กรุงเทพ** เพราะงั้นแปลว่าข้อมูลเราไม่เวิร์ค ทำให้มีคำว่า Exploration & Analysis จาก large quantity of data เพื่อค้นหารูปแบบที่มีความหมายและเป็นประโยชน์
- แต่ถ้าเราไปคนเจออะไรที่รู้อยู่แล้ว เป็น fact อยู่แล้วก็ไม่นับว่าเป็น knowledge

KDD

- selection > preprocessing > transformation > data mining > interpretation/evaluation
- เช่นเวลาขายของ บนเน็ต เราซื้ออย่างนึง อีกอย่างจะโผล่มาให้กดซื้อด้วย

Evaluate of Database Technology

- เกิดมานานแล้วจ้า
- เดี๋ยวนี้ก็ไปทำ warehouse, mining

Database VS Data Mining

- Database จะเป็นพวกการ search เช่น คนทำบัตรเครดิต ใครชื่อ Smith บ้าง เดือนที่แล้วใครยอดซื้อสูงกว่า 100 บาทบ้าง?
- Data Mining จะเป็นการดูว่า ใครบ้างที่มาทำบัตรเครดิต แล้วจะสร้างปัญหา, ใครลงเรียนวิชานี้แล้วจะได้เอบ้าง? เอาลักษณะ เอาโมเดลมาวิเคราะห์ ,ถ้ามีฐานข้อมูลลูกค้า แบ่งลูกค้าได้เป็นกี่กลุ่มที่มีพฤติกรรมผู้บริโภคที่ใกล้เคียงกัน
- การทำ Mining ไม่ใช่การดูจาก group by เหมือนพวก Database โจทย์ของสองแบบต่างกัน
- Mining จะเป็นการที่ข้อมูลพาเราไปเจอความรู้ข้อมูลใหม่อะไรบางอย่าง

Data, Information, Knowledge

- Data เป็นข้อมูลดิบที่สร้างมาจากระบบต่างๆ เช่น เว็บ ระบบเก็บข้อมูลต่างๆ
- Information เป็นพวกรายงาน ที่ได้มาจาก Warehouse เช่น กราฟต่างๆ
- Knowledge เป็นการสกัดข้อมูลมาจาก Information ที่เป็นความรู้ใหม่ๆ Guide สำหรับการตัดสินใจ

Database VS Data Warehousing

- Warehouse อยากรู้ว่ายอดขายของจังหวัดนี้เป็นเท่าไหร่ ที่อำเภอนี้เป็นยังไงบ้าง ,hierarchy มาเกี่ยวข้อง
- Database อยากรู้ว่ายอดขายเดือนที่แล้วเป็นเท่าไหร่
- Mining เป็นการดูว่า เดือนหน้ายอดขายจะเป็นเท่าไหร่ แล้วทำไมมันถึงเป็นแบบนั้น จะขึ้นหรือจะลง ดูแนวโน้ม ใช้ Algorithm ใช้คอมพิวเตอร์ที่มีสมรรถนะเข้ามาช่วย

Search VS Discovery

- Mining จะเป็นการ Discover ไม่ใช่การค้นหาธรรมดา ถ้ามีโครงสร้างจะเป็นแบบ graph-mining, sequence-mining ถ้าไม่มีโครงสร้างจะเป็น text-mining, web-mining, image-mining
- Search จะเป็นพวกค้นหาเฉยๆ เช่น ถ้ามีโครงสร้าง จะเป็น Data querying, warehousing ถ้าไม่มีโครงสร้างก็จะเป็น information retrieval

KDD Process

- Selection: เป็นการเลือกข้อมูล มาจากข้อมูลทั้งหมดเพื่อเอามาทำ mining เช่น จะดูเกรดนักเรียน ก็ไม่ต้องเอาอาชีพของพ่อแม่มาดูก็ได้ เดี๋ยวนี้มีวิธีในการเลือกที่ง่ายๆ โดยเป็นการเลือก attribute ที่ส่งผลต่อ target (Feature Selection Algorithm) เพราะบางที เป็นพันๆ Attribute จะเลือกยังไง ข้อมูลที่เป็น text จะเลือกคำยังไง
- Preprocess: Data quality, sampling, clean data,… เยอะ เป็นส่วนที่กินเวลามากที่สุด
- Transform: Algorithm จะต้องการ specific format เราก็ต้องเปลี่ยนให้มันตรงกับที่ต้องการ เช่น อัลกอต้องการข้อมูลที่มีค่าไม่เกิน 1 เราก็ต้องทำ normalization ก่อน
- Data mining: เลือกวิธีที่จะใช้ ต้องเลือกเทคนิคที่เหมาะสม ทำ clustering? classification? มีเทคนิคให้เลือกเยอะ เราต้องจูนพารามิเตอร์ของเทคนิคที่จะใช้ให้ดี เพื่อให้ได้โมเดลที่ดีที่สุด
- Interpretation: เป็นการตีความ เพื่อให้ได้ output pattern ที่ถูกต้องและน่าสนใจ ถึงจะได้ knowledge ออกมา

Data Mining

- เกิดจากไอเดียของพวก ML / AI / pattern recognition / statistic / database system เพราะพวกนี้มีเทคนิคที่ต่างกันไป รองรับต่างกันไป
- วิธีเฉพาะตัวบางอย่าง อาจจะไม่เหมาะเช่น ข้อมูลขนาดใหญ่ มี dimension เยอะ ข้อมูลมาจากหลาย source ทำให้ mining จะเอาวิธีพวกนี้ มาทำให้เหมาะกับสเกลใหญ่ๆ

Statistics -> KDD

- Stat เข้าใจยาก ดึงมาส่วนนึงแล้วบอกว่าอธิบายได้ทั้งหมด แต่ Mining เน้นความแม่นยำ มากกว่า ทำให้กระบวนการ Stat สามารถตีความได้ง่ายขึ้น แล้วนักstat ก็มีน้อย (สมัยก่อน) ใช้คนทำ Mining ก็ทำได้

Basic Data Mining Task

- Prediction modeling เป็นการทำนาย ที่ไม่ใช่ตัวเลข เช่น ทำนายว่า จะอยู่หรือไป หุ้นขึ้นหรือลง(classification) แต่ถ้าเป็นตัวเลขก็จะเป็นพวก ทำนายเกรด (prediction)
- Clustering แบ่งข้อมูลเป็นกลุ่มๆ ที่แต่ละกลุ่มมี record ที่เหมือนกันหรือคล้ายกัน แบ่งกลุ่มอัตโนมัติ ลักษณะวิธีการดูว่าของสองอย่างเหมือนกัน ดูยังไง เช่น โมเดล K-mean
- Link Analysis หา pattern ที่เกิดขึ้นด้วยกันบ่อยๆ เช่นพวก ซื้อAจะซื้อBด้วย
- Summarization ให้เอกสารไปแล้วทำการย่อ เป็นการอธิบาย Data สร้าง abstract
- Time Series Analysis เช่น ดูพวกหุ้นตามเวลา

Classification: Application

- Direct Marketing เช่น การดูว่า เวลาจะส่งบัตรเชิญไปหาใคร ที่มีแนวโน้มว่าจะทำการซื้อ ทำให้ไม่ต้องส่งไปหาทุกคน
- Churn Prediction เป็นการดูว่า ลูกค้าคนไหนจะหนีจากเราไป เราก็ส่งโปรโมชั่นไปให้ลูกค้าก่อนที่สายเกินไป

Clustering Definition

- แบ่งกลุ่มให้แล้ว เอามา visualize ให้เราดูได้ด้วยว่า ข้อมูลแต่ละกลุ่มเป็นยังไง เรามีความสนใจในกลุ่มไหน

Sequential Pattern Discovery: Examples

- เช่น ซื้อหนังสือเล่มนี้ ต่อด้วยเล่มนี้ ...

www.kdnuggets.com

หนังสือใช้แค่ครึ่งเทอม แค่บางบท ซีลอกเอาก็ได้

Data Mining -2

- จากควิซ ค่าบางค่า correlate กันเกินไป ทำให้ไม่จำเป็นต้องทำ Data mining ก็ได้ เพราะฉะนั้น ค่านั้นเลยไม่เหมาะสม

Supervised Learning

- การเรียนรู้แบบมีตัวอย่าง
- มีตัวอย่าง มี target ที่ชัดเจน ใช้ training เป็น input ที่มีคำตอบระบุไว้อยู่แล้ว
- ค่าที่ระบุ อาจจะเป็น Mathematical หรือ Categorical เช่นพวก เพศ ค่าสูงหรือต่ำ แล้วก็เป็นค่า predict ที่ระบุตัวอย่างคำตอบไว้แล้ว
- ทำ supervised เพราะเราต้องการทำนาย input ที่เข้ามาใหม่ ว่าจะได้คำตอบเป็นอะไร
- ถ้าเราไม่มี objective ชัดเจนเราจะไม่รู้ว่าอะไรจะเป็น target ของการทำนายนี้ ทำให้ไม่สามารถทำได้หรือ รู้แต่ว่าไม่มีค่าคำตอบของข้อมูลตัวอย่าง
- เพราะฉะนั้นปัญหาคือ ไม่มีคำตอบ(เฉลย) ไม่รู้ว่าจะใช้อะไรเป็น feature ในการทำนาย
- Classification/prediction

Supervised: Classification

- ทำนาย categorical target เช่น ซื้อหรือไม่ซื้อ
- การพิจารณา case เช่น การทำนายลูกค้าหนึ่งคน ต้องพิจารณาจากการโทรศัพท์มากกว่าหนึ่งครั้ง ไม่ใช่แค่ transaction การโทรครั้งเดียว เพราะฉะนั้น grain ของการทำนายนี้คือ ลูกค้า (Customer ID)
- ตัวแปรที่เอามาใช้ในการทำนาย เรียกว่า predictor, feature, attribute ส่วนค่าที่ต้องการคือ outcome (ส่วนใหญ่จะเป็นค่า binary ใช่หรือไม่ใช่ แต่ multiclass ก็มี target มากกว่า2)

Supervised: Prediction

- จะเป็นการทำนายตัวเลข

Unsupervised Learning

- ต้องการแบ่งข้อมูล ไม่มี target
- ใช้การทำ clustering, association rules, data exploration, data visualization
- อย่างเวลา clustering จะแบ่งเป็นกลุ่มของข้อมูล แล้วแสดงรายละเอียดให้เราเห็นเลยว่าแต่ละกลุ่มของข้อมูลเป็นแบบไหน
- Association Rule จะสร้างกฎออกมาว่า อะไรจะเกินขึ้นกับอะไร เช่น ลูกค้าซื้อนมจะซื้อโค้กด้วย
- Data Exploration จะแสดงข้อมูลเยอะๆ ด้วยเทคนิค visualization แบ่งข้อมูลเป็นหน่วยๆ ไม่เหมือนเป็นกลุ่มแบบ clustering
- Data Visualization

Steps in Data Mining

- กำหนด purpose ที่ชัดเจนของการทำ mining ทำ supervised, unsupervised
- ไปเอา Data มา ถ้ามันเยอะมาก จะทำ sampling ก่อนมั้ย โดย sampling ต้องทำแบบมีหลักการ ให้ไม่เสียการกระจายของข้อมูล ถ้าข้อมูลน้อย อาจจะต้อ งgenเพิ่มเองด้วย

- Explore, clean, pre-process
- Reduce attribute บางอันอาจจะเยอะไป เกิด correlation มากเกินไป ต้องเอาออก และถ้าเกิดว่าเราทำ Supervised เราต้องแบ่งข้อมูลเป็นส่วนๆ ที่จะเอาไว้ทำ training, testing, validation
- เลือกเทคนิคที่ใช้ว่าจะทำอะไร classification, clustering, ….
- เลือกเทคนิคที่ใช้ อัลกอริทึมของการทำงาน เพื่อให้ได้โมเดลที่ดีที่สุด
- ถ้าเกิดว่ายังไม่ดี ก็ต้องกลับไปทำใหม่ ข้อมูลยังไม่ดีก็แก้เพิ่ม ทำการจูนระบบ
- ต้องมีการทำการ compare model เช่น F-measure ,Precision , Recall หรืออาจจะเลือกหลายๆ อัลกอริทึม
- <u>โมเดลที่ดีที่สุด คือ โมเดลที่แม่นที่สุด แม่นกับ Unseen ไม่ใช่แม่นกับ data ที่มาเทรน</u>

Obtain Data: Sampling

- ปกติแล้วทำกับข้อมูลขนาดใหญ่ ทำให้บาง tools อาจจะไม่รองรับ
- ข้อมูลบางอย่างอาจจะไม่ได้สนใจจริงๆ เช่น อย่างคนเป็นมะเร็ง จำนวนปริมาณข้อมูลจะน้อยมากๆ ส่วนใหญ่จะเป็นคนที่ไม่เป็น ทำให้ข้อมูลมันน้อยเกินไป ต้องทำ oversampling ทำให้คลาสของคนที่เป็นมะเร็งมีจำนวนมากขึ้น ทำให้ข้อมูลสองส่วน balance กัน แต่การ oversampling ก็ต้องทำให้เหมาะสมด้วย ไม่ให้มันเว่อร์เกินไป
- แล้วเวลามีการทำ sampling เราก็ต้องทำการ adjust result ด้วย

Preprocessing

- กินเวลามาก 60-70% ทั้งหมด เพราะเวลาใช้เทคนิค มันมีอัลกอออยู่แล้วแค่ทำการปรับให้มันแม่น
- แต่ข้อมูลที่เราเอามา ส่วนใหญ่จะไม่ดี ต้องเอามาแก้ไข แปลง ไม่งั้นโมเดลเราจะออกมาไม่มีประสิทธิภาพ
- หลักๆ เช่น
- แปลงข้อมูลจากตัวเลข เป็นประเภท หรืออย่างอื่น บางอัลกอเหมาะกับข้อมูลตัวเลข บางอันเหมาะกับแบบประเภท บางอันต้องการค่า 0-1 เท่านั้น

Variable Handling

- สำหรับแบบ Numerical บาง tool ไม่รองรับข้อมูลแบบละเอียด เราต้องทำ discretize ให้เป็นช่วงก่อน
- ถ้าเป็น categorical ก็เช่น Naïve Bayes หรือบาง tool เป็น binary ว่าใช่หรือไม่ แทนที่จะใส่ค่าได้ว่า ยี่ห้อ A B C เราต้องแก้เป็น ใช่ A,ไม่ใช่A ใช่B,ไม่ใช่B

Detecting Outlier

- บางข้อมูลเป็นค่าที่แตกต่างจากค่าอื่นๆ มากๆ ซึ่งจะเป็นค่าที่ส่งผลทำให้โมเดลเพี้ยนไป เช่น คนอื่นเงินเดือนประมาณสามหมื่น แต่มีสองสามคนสามล้าน ทำให้โมเดลมันเพี้ยน
- อาจจะต้องใช้ domain knowledge เพื่อช่วยเอามันออกไป

Handling Missing Data

- บางอันจะถูกดรอปไปเพราะอัลกอริทึมเห็นว่าค่าหลายๆค่า หายไป
- แบบแรก omission เราอาจจะตัดทิ้งไปเลยได้ ถ้าเกิดว่าจำนวน row มันไม่เยอะมาก แต่ถ้าเกิดว่ามันมีจำนวนมากๆ เราก็ไม่ควรจะเอามันทิ้ง

- แบบที่สอง ถ้าเกิดว่ามีจำนวน row มาก เราก็ replace อาจจะ replace โดยการเดาค่าจาก mining ก่อนก็ได้ หรือจะแทนด้วยค่า min/max

Data Reduction

- ลดจำนวนข้อมูลให้น้อยลง
- ลด column ลดจำนวนตัวแปรให้น้อยลง
- ลดจำนวน record ด้วยการ clustering ทำให้เราได้ตัวแทนของข้อมูลมา มาใช้แทนข้อมูลทั้งหมดในกลุ่ม

Normalization

- การทำให้ข้อมูลอยู่ในสเกลเดียวกัน

Partitioning the Data

- แบ่งข้อมูลเป็นสองส่วน คือ Training , Validation
- Training คือเอาไว้เทรน มีคำตอบอยู่แล้ว
- Validation คือข้อมูลที่มีคำตอบเหมือนกัน เอาไว้ส่งให้อัลกอเพื่อปรับแต่งโมเดล ว่าจะทำงานกับ Unseen ได้มั้ย **ใช้ในการแก้ปัญหาเรื่อง Overfitting** (ข้อมูล fit กับโมเดลมากเกินไป ทำให้เวลาทำงานกับ Unseen จะตอบไม่ได้)
- ส่วนการ Testing คือข้อมูลที่เอาไว้ใช้ตรวจสอบดูเลยว่าจริงๆ แล้วโมเดลเราแม่นมากน้อยแค่ไหน มีเฉลยเหมือนกัน
- การ Test จะเป็นส่วนสำคัญที่ใช้ในการเลือกโมเดล เราต้องดูว่าทดสอบแล้วได้ความแม่นมากที่สุด ก็จะเลือกอันนั้น

Problem of Overfitting

- อย่างพวก stat จะเน้นการหาโมเดลที่ตรงกับข้อมูลมากที่สุด ซึ่ง Data Mining จะไม่ต้องการแบบนั้น เพราะต้องการโมเดลที่เหมาะกับ unseen
- เราต้องลองกับหลายๆ โมเดลดูว่าแบบไหนจะดี เหมาะกับข้อมูลใหม่ อย่าง decision tree ถ้าเกิดว่า tree มันลึกมากๆ แปลว่ามันเกาะติดกับข้อมูลที่เอามาเทรนมากๆ ทำให้ไม่เหมาะกับข้อมูลใหม่ๆ ซึ่งถ้าเราตัดกิ่งมันทิ้งไป ก็จะทำให้โมเดลเรา General มากขึ้น

Association Rules – Lecture 3

Association Rules

- เป็น unsupervised learning
- ศึกษาว่า อะไรจะไปคู่กับอะไร เช่น ซื้อของ A แล้วจะซื้อของ B , เป็นอาการ A แล้วจะเป็นโรค B
- บางทีเรียกว่า Marker Basket Analysis, Affinity Analysis
- ข้อมูลนำเข้าจะเป็น set ของข้อมูล ในแต่ละ transaction อาจจะไม่ต้องเป็นข้อมูลระดับ นมยี่ห้อxxx เราแค่จับมันเป็นกลุ่มว่านม ก็พอ
- k-item set คือ itemset ที่มีข้อมูล k ตัว
- Support count (σ) เป็นความถี่ของการเกิดของ itemset
- Support คือ ค่าที่บอกว่าเกิดเท่าไหร่ เช่น {milk,bread} = 2/5
- Frequent itemset คือ itemset ที่มี support มากกว่าหรือเท่ากับค่า minsup threshold

Definition: Association Rule

- Association Rule จะเรียกต่างกันเช่น {Milk, Diaper} -> {Beer} อาจจะเรียกว่า head, body/ antecedent, consequence/ left-hand, right-hand
- ปกติเราได้กฎออกมาเยอะมาก แต่จริงๆ เราอยากได้แค่กฎที่น่าสนใจ strong rules/ interesting rules
- Rule evaluation metrics
    - Support (s): ว่าจำนวนการเกิดมีความถี่เท่าไหร่
    - Confident (c): วัดใน transaction ที่มี Y มีจำนวนเท่าไหร่ที่เกิด X ด้วย
- เช่น กฎว่า {x, y} -> {z}
- c= $\frac{\sigma(\{x,y,z\})}{\sigma(\{x,y\})}$
- บางที การวัดว่ามันเกิดร่วมกัน เกือบ 100% ก็อาจจะไม่มีประโยชน์ เพราะว่าจำนวนการเกิดมันน้อย แม้ว่าจะเกิดร่วมกันมาก ก็ไม่น่าสนใจ
- เราเลยต้องใช้ค่า support เข้ามาช่วย เลยต้องใช้ทั้งค่า confident และ support ในการพิจารณา

Association Rule Mining Task

- วิธีถ้าเกิดว่าเราจะทำเอง จะทำยังไง -> Brute Force? แล้วต้องมานั่งคำนวณค่า support/confidence
- เกิดวิธีช่วยให้หากฎได้เร็วขึ้น กฎที่น่าสนใจ จะมาจาก Frequent itemset นับค่า support มาก่อน (Rule ที่มาจาก itemset เดียวกัน เช่น {milk, diaper} -> {beer} กับ {milk, beer} -> {diaper} จะมีค่า support เท่ากัน แต่ค่า c ของกฎจะต่างกัน)

Generating Association Rules

- ขั้นแรกไปหา frequent itemset ก่อน (ได้ค่า support) >> กินเวลามากกว่า เพราะว่าต้องหาหลาย combination แล้วนับ ต้องไปอ่าน HDD มา ต้องหาวิธี optimize

- ต่อไปเอาแต่ละ itemset เอาออกมากระจายหากฎ ดูค่า confident แล้วเลือกอันที่ผ่าน threshold ออกมา เร็วกว่าเพราะนับจาก Memory เลย

Frequent Itemset Generation Strategies

- ลดจำนวน candidates โดยใช้การ pruning เพราะบางทียังไงมันก็อาจจะไม่ผ่านค่า support อยู่แล้ว
- ลดจำนวน transaction แทนที่จะมอง database เป็น transaction เราก็มองกลับกัน เช่น จากมองว่า transaction 1 -> bread, milk เราก็มองเป็น bread อยู่ใน transaction ไหนบ้าง
- ลดจำนวนของการเปรียบเทียบ ใช้ data structure ไม่ต้องแมพทุกๆ candidate กับ ทุกๆ transaction

Mining Association Rules

- Apriori เป็นอัลกอริทึม คือ
- ถ้า superset มัน frequent subset ก็จะ frequent ด้วย เช่น BCE ใช่ BE ก็ต้องใช่ด้วย
- ถ้า subset ไม่ frequent superset ก็จะไม่ด้วย

Multi-dimension association rules

- ถ้าเกิดว่าข้อมูลนำเข้าไม่ได้เป็น binary database แต่เป็น relational database ก็จะสร้างกฎได้เหมือนกัน

Lift

- เป็นค่า ratio ที่แสดงค่าความขึ้นต่อกันของทางซ้ายของกฎและขวาของกฎ ว่าแปรผันต่อกันจริงๆ หรือเปล่า
- ถ้า lift >1 ทางซ้ายส่งผลทางบวกต่อทางขวา
- ถ้า lift =0 ไม่ขึ้นต่อกัน
- lift <0 ทางซ้ายส่งผลต่อทางขวาเชิงลบ
- คือค่า correlation นั่นเอง correlation = Confident/Expected Confident (support ของ ทางขวา)

K-nearest neighbor - Lecture 4

(Supervised Learning)

Data Classification

- ต้องมี training data set
- training data set จะมี set ของ attribute ต้องมีส่วนที่บอกประเภทด้วย
- เราต้องการหาโมเดลสำหรับ attribute ต่างๆ และคลาส
- เพื่อทำนาย, กำหนดคลาสให้กับ unseen record
- data ที่จะเทรน ก็ต้องมีครบแบบ เช่น ทำนายว่าลูกค้าจะอยู่หรือไป ก็ต้อง อยู่/ไป
- ต้องมี validation set (เอาไว้ปรับโมเดล) , test set (ประเมิน accuracy)
- confusion matrix จะเกิดจากการ train / test ก็ได้ แต่ผลจะออกมาต่างกัน
- ข้อเสียของงานแบบนี้คือ ต้องมีคลาสมาก่อน คลาสอาจจะไม่ชัดเจนด้วย
- คลาสยิ่งมาก ความแม่นจะน้อยลง
- ได้กฎออกมา ทางซ้ายจะเป็นคลาสเสมอ ได้ classification rules
- ถ้า attribute ข้อมูลเยอะ ก็อาจจะต้องลดลง
- ลักษณะการใช้ข้อมูลเทรน พวก validation มาช่วยปรับโมเดล โดยสร้างโมเดลไว้ก่อน เรียกว่า eager learner
- ถ้าเป็น แบบไม่สร้างโมเดล เราเอา unseen มาเทียบเลย จะเรียกว่า lazy learning (instant base)
- แบบ lazy เวลามีข้อมูลใหม่เข้ามาเสียเวลา cost เพราะไม่มีโมเดลไว้ก่อน
- k-nearest จะเป็นแบบ lazy

K-nearest

- เวลามี data ใหม่เข้ามา จะไปดูว่า ในข้อมูลเดิม มีอันไหนคล้ายมันมากๆ บ้าง (เรียกว่า neighbor)
- neighbor ส่วนใหญ่อยู่คลาสไหน เราก็จะตอบคลาสนั้น
- k คือ เลขจำนวน neighbor ที่จะไปดึงมาพิจารณา
- เวลาจะหาความเหมือน ก็ต้องใช้ distance metric อย่างครั้งก่อนๆ ก็พวก Euclidean distance หรือจะเลือกแบบอื่น แล้วก็กำหนดค่า k แล้วเวลามี data ใหม่มา ก็ไปคำนวณทุกๆ distance แล้วก็เลือก record มาจำนวน k
- การที่ทำนายออกมาไม่ดี อาจจะเป็นเพราะเลือก distance function ไม่ดี ค่า k ไม่เหมาะสม หรือว่าเป็นเรื่องของจำนวน attribute
- k น้อย เช่น k =1 -> overfitting เพราะว่า record นั้นอาจจะเป็น noise ไม่ใช่ข้อมูลจริงๆ ก็ได้ แค่มันบังเอิญตรง, k=n เยอะมาก ก็ไม่ต้องทำแล้ว มันจะ error เยอะมาก เลือกเอาจากคลาสที่เยอะที่สุดเลยง่ายกว่า
- เราจะพยายามเลือก k ต่ำที่สุดที่ได้ accuracy สูงสุด
- k มากพอ จะทำให้เรากำจัดส่วน error เพราะ noise ได้K

K-NN for prediction (numerical outcome)

- เอาพวก weight เอา average ออกมา คำนวณได้ค่าที่เป็นตัวเลขออกมา

K- nearest problem

- scale problem
- การกระจายของหน่วยวัดที่มีช่วงที่ต่างกันมากๆ เช่น salary , age, height
- จำนวนตัวแปรเยอะ training set ใหญ่ ก็จะใช้เวลา เสียเวลาเพิ่มเป็น expo
- ยิ่งตัวแปรมาก distance ยิ่งไกล้ (curse of dimension)

K-nearest advantage

- ง่าย
- ไม่ต้องมีการกำหนดว่าจะต้องมีการกระจายข้อมูลเท่าไหร่ (เวิร์คทุกกรณี)
- ใช้งานได้ ไม่ต้องสนใจเรื่องสถิติ ว่าตัวแปรขึ้นต่อกันไม่ขึ้นต่อกัน ไม่สนใจ

Evaluation

- Error คือ แยกประเภทออกมาได้ไม่ตรงกับคลาสที่ถูกต้องจริงๆ
- Error rate คือวัดเปอร์เซ็นต์ที่แยกออกมาถูกต้อง

Naïve Rule

- คล้ายกับ majority vote ส่วนใหญ่เราจะใช้ most prevalent class มาตอบ
- ส่วนใหญ่จะใช้เป็น benchmark แล้วออกกราฟมาเทียบกับกราฟของโมเดล

Separation of Records

- ถ้าเกิดว่าเราเลือก feature ที่ดีจะทำให้เราแยกได้ดี -> high separation of records ซึ่ง record แต่ละคลาส แยกกันได้ดี ทำให้ low error
- แต่ถ้าเราเลือกใช้ feature ที่ไม่ดี -> low separation of records แต่ละ record แยกกันไม่ดี ทำให้ error มากกว่า อาจจะไม่ค่อย improve จาก naïve rule เท่าไหร่
- อย่าง decision tree มันจะเลือก feature ให้เราเองเลยใน algorithm

Confusion Matrix

- ส่วนใหญ่เราจะเอา precision/recall มาพิจารณา ว่าทั้งสองอย่างต้องสูง
- ดู F-measure คือดูทั้ง precision/recall ไปพร้อมกัน
- บางทีจะดูเป็น TP: true positive, TN: true negative, FP: false positive, FN: false negative (true,false คือ ทำนายถูกผิด) True Positive: ทำนายถูก ว่ามันตอบใช่ FP: ทำนายผิดว่ามันตอบใช่
- ส่วนใหญ่เราจะต้องการลด FN

Cutoff for classification

- ส่วนใหญ่โมเดลจะใช้ prob ในการตัดสินใจว่าจะเลือกโมเดลไหน
- default = 50% เราอาจจะกำหนดเพิ่ม/ลดได้ แต่ที่วัดมา เค้าบอกว่า 50% ดีที่สุดแล้ว
- น้อยเกินมากเกินอาจจะทำให้ความถูกต้องมันลดลง

When one class is more important

- บางทีเราต้องการคลาสบางคลาสมากๆ เราอาจจะต้องยอมให้ error rate โดยรวมมันเพิ่มขึ้น แต่ว่าคืนค่า TP มามากขึ้น

Alternate Accuracy Measure

- Sensitivity: % of C1 class correctly classified
- Specificity: % of C0 class correctly classified
- ROC curve: กราฟพลอตระหว่าง ความแม่นคลาส positive – ความแม่นคลาส negative ต้องการกราฟที่มีพท.ใต้กราฟมาก

Lift and Decile Charts: Goal

- เช่น เวลาเราจะต้องการคำตอบลูกค้า ส่งจม.ไป แล้วลูกค้าจะตอบ เราอยากให้โมเดลช่วยเราลด cost คือ ส่งไปน้อยๆ แต่ตอบกลับมา yes เยอะ
- Lift chart: cumulative performance

Decision & Regression Trees

Trees and Rules

- Rule base
- Classification rule คือ ด้านซ้ายจะเป็น condition ด้านขวาจะเป็น class attribute
- Decision Tree: Categorical
- Regression Tree: Numerical
- Tree จะเป็นกฎว่า ถ้ามี condition เป็นแบบนี้ จะได้ผลเป็นแบบไหน
- ยิ่ง attribute มีค่าได้เยอะ ก็จะทำให้ กิ่ง ของ tree มีจำนวนเยอะด้วย
- leave node จะเป็น class label
- Training หนึ่งอัน สามารถสร้างได้หลาย tree, model ที่เข้าได้กับ data
- เราก็เอาแต่ละ tree ไปวัด เช่น confusion matrix, ROC curve แล้วเลือกอันที่มันแม่นกับ unseen
- 1 node ก็คือ subset ของ training

Key Ideas

- ขั้นแรกในการสร้าง คือ recursive partitioning แบ่งๆ ไปเรื่อยๆ จนท้ายสุด จะประกอบไปด้วย record ที่เป็นคลาส 0 ทั้งหมด / 1 ทั้งหมด

Hunt's Algorithm

Pre Pruning

- ถ้าเกิดว่าเราเจอจุดที่มันจะ overfit ก็หยุดก่อน

Data Clustering- Lecture 5

- Clustering = a set of Clusters

Cluster Analysis

- หากลุ่มของ object ที่ไม่ต้องมี label ก็ได้ ดูว่าแบ่งออกมาได้ยังไง
- ภายในแต่ละกลุ่ม distance จะใกล้กัน (intra-cluster distance) ถ้าเป็นระหว่างกลุ่ม จะเรียกว่า inter-cluster distance
- เวลามีข้อมูลมา ก็ลองเอามาทำ clustering ก่อน ก็จะรู้ว่าข้อมูลมีกลุ่มไหนบ้าง
- บางคนบอกว่าเป็นการทำ preprocessing ไม่ใช่ mining เพราะบางทีทำ clustering เสร็จ ก็เอาแต่ละกลุ่มไปทำ Association rule สำหรับกลุ่มนั้นๆ ต่อ
- ตัวใหม่ที่วิ่งเข้ามา จะพิจารณาด้วย distance ว่าใกล้ไกลกลุ่มไหน แต่ละกลุ่มจะมีตัวแทนอยู่ตัวหนึ่ง (centroid)

Classification vs. Clustering

- อาจจะเอา classification data ที่มี label เข้าไปทำ clustering
- ถ้า clustering ได้ดีจริงๆ มันก็น่าจะแบ่งได้ label ที่ดีจริง (ดี = pure)

Examples of Clustering Application

- Marketing: แบ่งกลุ่มลูกค้า
- Astronomy: กลุ่มดาว
- อย่างเวลาการค้นหาเอกสาร เราอาจจะคืนค่าให้เป็นกลุ่มของเอกสารไปเลยก็ได้
- หรือทำ visualization ก็ได้

Note:

- ถ้าข้อมูลที่เอาไปทำ clustering มีข้อมูลเป็นตัวเลข จะทำให้สามารถวัด error rate ได้ วัด distance ได้ง่ายกว่า
- ถ้าเป็น categorical ก็จะมีวิธีการคำนวณที่ต่างออกไป
- เราจะต้องมี visualization ที่ดี จะทำให้เราสามารถแบ่งข้อมูลได้ดี อ่านข้อมูลได้ง่าย ถ้ามองจากกราฟ x,y ก็จะเห็นแค่สองมิติเท่านั้น พอเรามองแล้ว ก็จะได้ label ออกมา (เพราะตอนแรกยังไม่มี label มาให้)
- เพราะฉะนั้นต้องใช้ software ที่มีกราฟฟิกดีๆ

Algorithm

- มีหลายอัน บางอันก็จะเหมาะกับข้อมูลที่เป็นตัวเลข บางอันก็เหมาะกับที่เป็นประเภท
- การที่เราถามว่า unseen นี้จะไปอยู่ใน cluster ก็จะได้ค่า prob มาด้วยว่า มีโอกาสจะอยู่ในกลุ่มนี้แค่ไหน แต่บางอันก็ไม่บอก
- บาง clustering อาจจะทำแบบทั้ง exclusive/overlap อยู่ได้แค่ cluster เดียว หรือว่าอยู่ในหลาย cluster ก็ได้
- มีทั้งแบบ flat/hierarchical คือ แบ่งไปเลยชั้นเดียวหรือ มี cluster ใหญ่ครอบอยู่อีกที เป็นลำดับชั้นลงไป (partitional clustering – hierarchical clustering
- top down คือ มองว่าทั้งหมดเป็นกลุ่มใหญ่ก่อน แล้วค่อยแยกออกมา - bottom up มองว่าแต่ละอันเป็นคนละกลุ่มก่อน แล้วค่อยๆ รวมกันขึ้นไป (ส่วนใหญ่จะเป็น hierarchical ด้วย)

Distance

- มีวิธีคำนวณหลายแบบ เช่นเอามาบวกลบกันเลย ใช้ weight ใช้ Euclidean distance
- ถ้าเป็น nominal attribute ก็จะใช้ 1 = ต่าง 0 = เหมือน
- Euclidean Distance ฮิตที่สุด

Normalizing

- อาจจะมีปัญหาถ้าเกิดว่ามี measurement เยอะมากๆ ทำให้ค่าระยะทางมันไกลเกินไป
- ก็ใช้ z-score ก็ได้

K-means Z (partitioning)

- ใช้เฉพาะกับ numerical data เท่านั้น
- k คือจำนวน cluster
- เริ่มด้วยการเลือก k ที่ดี ว่าจะให้มีกี่ center แล้วหยิบมาเป็น center
- แล้วระบุข้อมูลอื่นๆ ให้ไปอยู่ใกล้ๆ ตัวที่หยิบมาแล้ว
- จากนั้นคำนวณ centroid ใหม่ จนกว่าจะหยุด
- หยุดเมื่อ centroid ไม่ขยับแล้ว หรือขยับน้อยมากๆ

Problem

- ไม่รู้ว่าจะเลือก k ยังไง (อาจจะเป็นว่าเรารู้ล่วงหน้าว่ามีกี่กลุ่ม ซึ่งไม่เสมอไป เราอาจจะต้องค่อยๆ เปลี่ยนค่า k ไปเรื่อยๆ แล้วค่อยดูว่าแบบไหนดี)
- ตอนเริ่มจะเลือกแรนดอมมายังไง จะใช้อะไรตัดสิน (เราอาจจะบอก software ได้ว่าจะเลือกแบบไหน แล้วแต่ domain knowledge)
- k-means เลยมีข้อเสียในกรณีเริ่มต้นแบบนี้
- การแบ่ง cluster เราอาจจะดูจากระยะห่างระหว่าง cluster-cluster ว่ามันห่างกันดีรึเปล่า
- อาจจะมีปัญหาเรื่องกลุ่มกราฟหน้าตาแปลกๆ มันจะตัดกลุ่มออกมาได้ไม่ดี (วิธีแก้ บางคนบอกว่าให้แยก เยอะๆ แล้วพอแยกเสร็จ ก็ค่อยเอามารวมกันให้เป็นกลุ่ม)

K-mediods

- โดยอัลกอริทึม ค่า center ก็คือค่า mean
- แต่บางที mean ก็ไม่เหมาะ ถ้าข้อมูลมันโดดมากๆ ก็ควรจะใช้ค่า median
- เป็นการหา representative ของ cluster
- ถ้าเกิดว่าข้อมูลใหญ่มากๆ จะใช้การ sampling

Preprocess

- ก่อนทำให้ normalize , eliminate outlier

Post-process

- ถ้าเรารวมมันเป็นกลุ่มใหญ่ไป อาจจะต้องเขียนการแยกให้มันแยกออกจากกันเพิ่มเติม

- ถ้าตอนแรกเราแยกไว้เยอะ อาจจะต้องทำให้มันมารวมกัน merge กัน

Problem with centroid problem

- เราอาจจะไปหาเทคนิคอื่นๆ เพื่อหาค่า k มาก่อน แล้วค่อยเอามาทำ k-means

Hierarchical Method

- Agglomerative Methods (bottom-up) แยกเดี่ยวก่อน
- Divisive(top-down) รวมใหญ่ก่อน
- ดีกว่า k-mean ตรงที่ไม่ต้องกำหนดค่า k
- แต่ว่าถ้าตอนแรกๆ แยก แล้วเรารวมกันขึ้นไป ก็จะ backtrack กลับมาไม่ได้ รวมแล้วรวมเลย

How to Find K

- อาจจะทำ hierarchical แล้วดูว่าจะแบ่งยังไงดี แล้วค่อยเลือก k

DBSCAN (density base –solved K-means)

- ทำ clustering โดยใช้ความหนาแน่น
- Core point: เป็นจุดที่มี neighbor เยอะๆ ถ้า core มันรวมกันได้ ก็จะ merge เป็นกลุ่มเดียวกัน
- Border point:
- Noise point:
- ใช้ความหนาแน่น ทำให้สามารถใช้ได้กับ shape ที่แปลกๆ ขนาดต่างๆ

Clustering Validation

- Cluster interpretation: วัดกันด้วยความหมายของ cluster ว่าสามารถตีความได้ดีหรือไม่ จะใช้ visualization tools ช่วย
- Cluster Stability:
- Cluster Separation ดู distance ของ cluster

Interpretation

- ใช้ stat มาช่วยอธิบาย
- เวลาทำ clustering แล้ว อาจจะเอาตัวแปรที่ไม่ได้ใช้ทำ clustering มาเป็นตัว visualize

Evaluating K-means Cluster

- ใช้ SSE ดู เป็นการดูความห่างของobject แต่ละ cluster โดยเราอยากได้อันที่มี SSE ต่ำๆ
- k มาก SSE จะต่ำลง
- แต่ว่า good clustering > k น้อย SSE ต่ำด้วย

Cluster Evaluating

- ทำแบบตอนแรก คือเรารู้ label แล้วเอามาแยกโดยใช้ clustering วัดดูว่าตรงมากตรงน้อยแค่ไหน

Cluster Validation

- วัดด้วย F-measure เลยก็ได้
- แต่วิธีที่ดีที่สุด คือดูด้วยตาคน