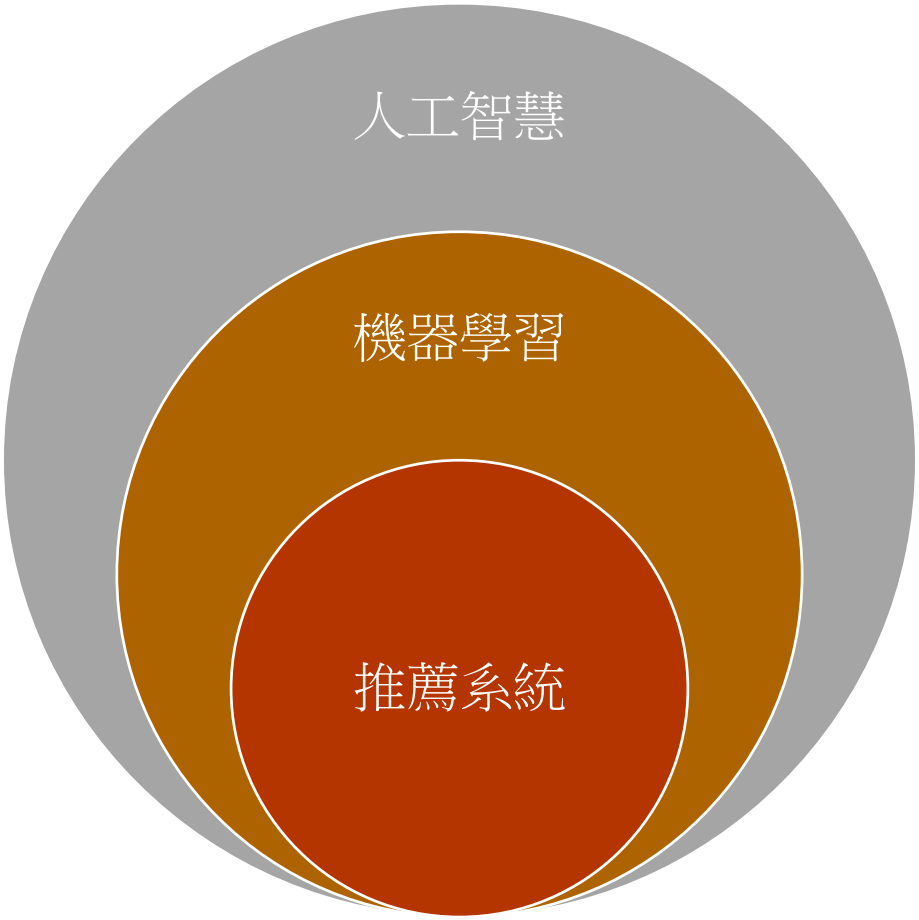
人工智慧、機器學習到推薦系統的應用

林守德
台大資工系教授
sdlin@csie.ntu.edu.tw

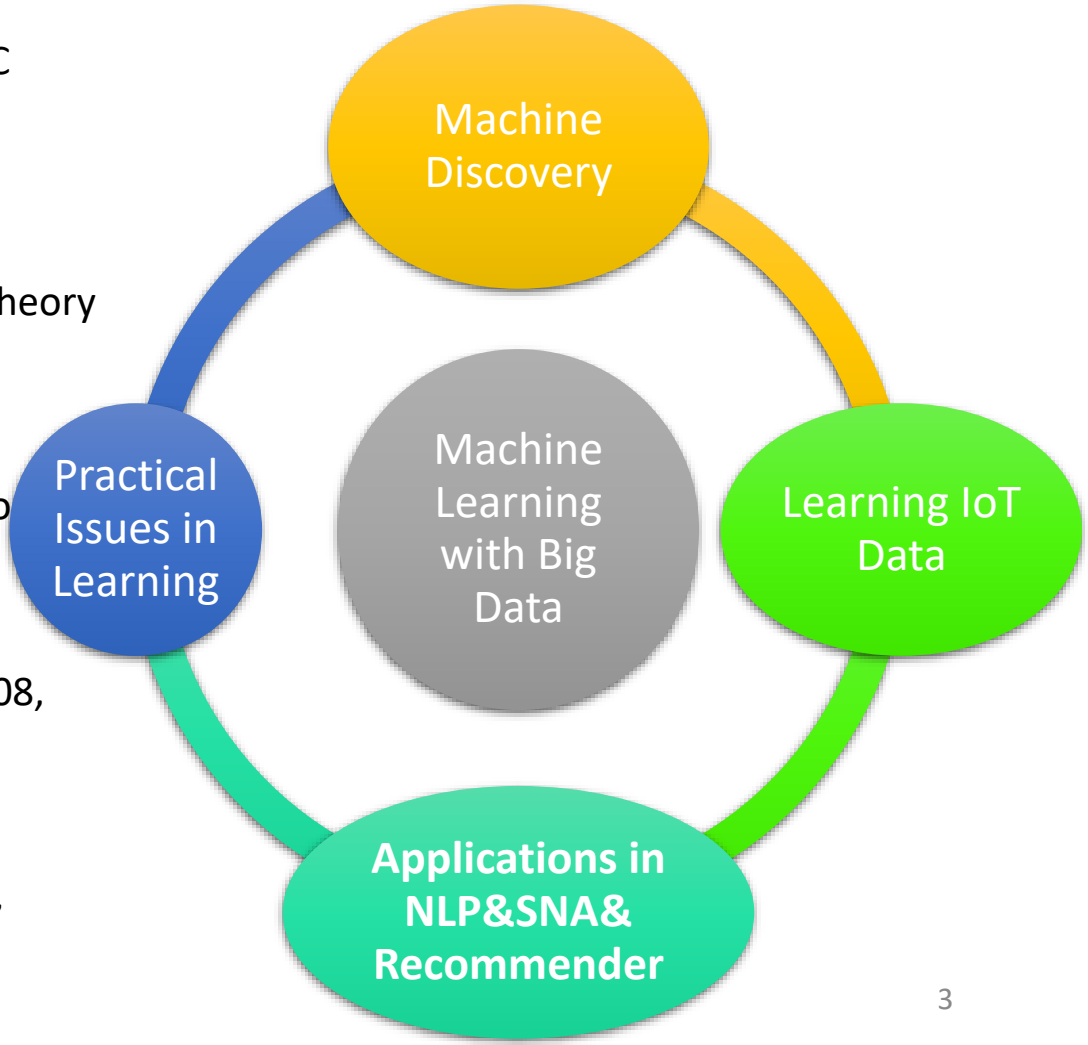


綜觀人工智慧

–人工智慧的過去，現在，和未來

About the Speaker

- **PI:** Shou-de Lin (machine discovery and social network mining lab)
  - B.S. in NTUEE
  - M.S. in EECS, UM
  - M.S. in Computational Linguistics, USC
  - Ph.D. in CS, USC
  - Postdoc in Los Alamos National Lab
- **Courses:**
  - Machine Learning and Data Mining- Theory and Practice
  - Machine Discovery
  - Social network Analysis
  - Technical Writing and Research Metho
  - Probabilistic Graphical Model
- **Awards:**
  - All-time ACM KDD Cup Champion (2008, 2010, 2011, 2012, 2013)
  - Google Research Award 2008
  - Microsoft Research Award 2009
  - Best Paper Award WI2003, TAAI 2010, ASONAM 2011, TAAI 2014
  - US Areospace AROAD Research Grant Award 2011, 2013, 2014, 2015, 2016



Agenda

- 人工智慧的定義與歷史
- 人工智慧現在的發展與應用
- 人工智慧的未來：從機器學習到機器發明

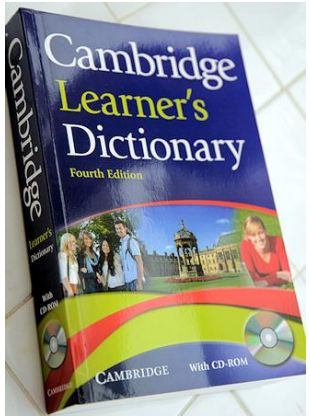
什麼是人工智慧（Artificial Intelligence）？

- 希望電腦能夠擁有智慧 （Strong AI)

- 希望電腦能夠展現出有智慧的外顯行為 (weak AI)

什麼是「智慧」？

智慧的定義



"The ability to learn, understand and make judgments or have opinions that are based on reason"

From AI researchers

- ". . . the ability to solve hard problems." M. Minsky
- in any real situation behavior appropriate to the ends of the system and adaptive to the demands of the environment can occur, within some limits of speed and complexity." A. Newell and H. A. Simon
- "Intelligence is the computational part of the ability to achieve goals in the world. Varying kinds and degrees of intelligence occur in people, many animals and some machines." J. McCarthy

從實務面上定義智慧

- 能夠觀察，瞭解，並對人事物進行反應（跟世界互動）
  - Reinforcement Learning, Markov Decision Process
- 能夠找到最佳的解決辦法
  - Optimization
- 能夠推論以及規劃
  - Inference and Planning
- 能夠學習以及調適
  - Machine Learning and adaption

人工智慧的歷史

- 千年來人類一直有著人工智慧的想像
  - 列子·湯問篇「越日偃師謁見王，王薦之，曰：「若與偕來者何人邪？」對曰：「臣之所造能倡者。」穆王驚視之，趣步俯仰，信人也。巧夫頷其頤，則歌合律；捧其手，則舞應節。千變萬化，惟意所適。王以為實人也，與盛姬內御並觀之。技將終，倡者瞬其目而招王之左右侍妾。」
  - Taros: 希臘神話中的金屬製機器人

人工智慧萌芽的三大起因

- 電腦的發明（ENIAC，1945）
- 圖靈測試(Turing test，1950)
- 達特毛斯第一屆AI會議（1955）
  - Participants including John McCarthy, Marvin Minsky, Claude Shannon, Allen Newell and Herbert A. Simon

1950~1960：人工智慧的啟蒙期

- Turing: Turing test
- Asimov: Three Laws
  - A robot may not injure a human being or, through inaction, allow a human being to come to harm.
  - A robot must obey the orders given it by human beings except where such orders would conflict with the First Law.
  - A robot must protect its own existence as long as such protection does not conflict with the First or Second Laws.
- First Chess Playing program (Strachey, Samuel)
- The first AI programming language: Lisp by McCarthy

1960~1970：獲得西方政府重視

- 美國政府提供大量的研究經費在AI
- 人工智慧前景一片看好
  - Herbert Simon: "machines will be capable, within twenty years, of doing any work a man can do".
  - Marvin Minsky "within a generation ... the problem of creating 'artificial intelligence' will substantially be solved".
- ELIZA（1966）: 第一個交談程式

1970～1980：AI的冬天

- AI的進展比期望中的慢很多
  - 需要搜尋的複雜度太高
  - 電腦encode人類的知識太慢
- Chinese Room Argument （Searle，1980）
  - Weak AI shows no intelligence

Funding Cut!! AI的冬天的來臨！

1980~1990：專家系統（Expert System)的崛起與沒落

- 專家系統：利用「規則」建立起類似專家的人工智慧系統
- 在80年代初期，許多專家系統開始商業化：
  - XCON（automatically select computer components based on user requirements）contains 2500 rules.
  - Saving $40 million dollars
- 然而，專家系統在80年代末期漸漸淡出工業界
  - 建置不易，需要領域專家
  - 無法通用
  - 規則可以融入各產品中

AI的第二個冬天！

1996年之後：AI再臨

- 多數學者放棄強人工智慧，往弱人工智慧靠攏
  - E.g. 搜尋引擎、語音辨識
- INTERNET的崛起，資料蒐集變得容易
- 電腦計算能力非比從前，很多過去無法實現的演算法變成可行

Afterwards, AI Research Are Mostly Driven By Competitions

Deep Blue （1996）

- 第一個贏過棋王的程式
  - 利用電腦平行運算能力從事地毯式的搜尋
  - 在演算法上面並沒有特別突破

Deep Blue
(photo taken by James at Computer History Museum)

DARPA Grand Challenge （2004）

- DARPA 提供一百萬美金給第一台橫越內華達沙漠的車
- 所有的隊伍在2004年都失敗了，但是在2005年有5台車成功橫越。
- 2007 進化為Urban challenge

Loebner Prize

- $25000 給予第一個被認為是人類的聊天機器人
  - 類似目的的系統如Siri, 小冰
- 頗具爭議性的AI競賽
- 至今仍未有人贏得最高獎金

RoboCup (1997-now)

- 機器人足球競賽
- "By the middle of the 21st century, a team of fully autonomous humanoid robot soccer players shall win a soccer game, complying with the official rules of FIFA, against the winner of the most recent World Cup.“

ACM KDD Cup (1997~now)

- It is an annual competition in the area of knowledge discovery and data mining
- Organized by ACM special interest group on KDD, started from 1997, now considered as the most prestigious data mining competition
- Competition lasts roughly 2-4 months

Team NTU's Performance on ACM KDD Cup

KDD Cups	2008	2009	2010	2011	2012	2013
Organizer	Siemens	Orange	PSLC Datashop	Yahoo!	Tencent	Microsoft
Topic	Breast Cancer Prediction	User Behavior Prediction	Learner Performance Prediction	Recommendation	Internet advertising (track 2)	Author-paper & Author name Identification
Data Type	Medical	Telcom	Education	Music	Search Engine Log	Academic Search Data
Challenge	Imbalance Data	Heterogeneous Data	Time-dependent instances	Large Scale Temporal + Taxonomy Info	Click through rate prediction	Alias in names
# of records	0.2M	0.1M	30M	300M	155M	250K Authors, 2.5M papers
# of teams	>200	>400	>100	>1000	>170	>700
Our Record	***Champion***	***3rd place***	***Champion***	***Champion***	***Champion***	***Champion***

IBM Watson （2011）

- 參與益智問答Jeopardy 節目打敗過去的冠軍
- Watson內部有上百個有智慧的模組（從語言分析一直到搜尋引擎）。
- 代表電腦在「知識問答」的任務已經能夠超過人類

AlphaGo (2016)

- Google’s AlphaGo 贏得號稱最困難的棋類：圍棋
- 與IBM的深藍不同，AlphaGo是結合硬體與複雜機器學習演算法

Agenda

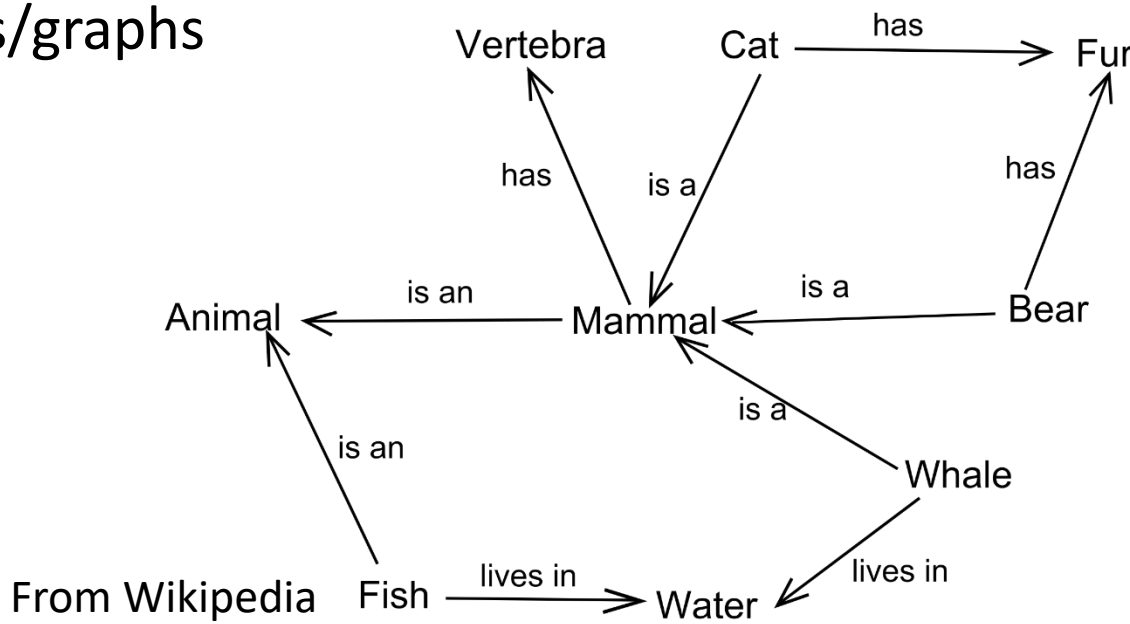
- 人工智慧的定義與歷史
- 人工智慧現在的發展與應用
- 人工智慧的未來：從機器學習到機器發明

實現人工智慧的技術重點

- 知識表徵 Knowledge Representation
- （機器）學習：（machine) Learning
- （機器）規劃 ：Planning
- 搜尋與最佳化 Search and Optimization

知識表徵 Knowledge Representation

- 知識表徵的方法有很多，取決於將來如何「使用」這個知識。
  - Logical Representations, e.g. ∀ x, King(x) ∩ Greedy (x) ➔ Evil (x)
  - Rules (can be written in logic or other form)
  - Semantic Networks/graphs
  - Frames

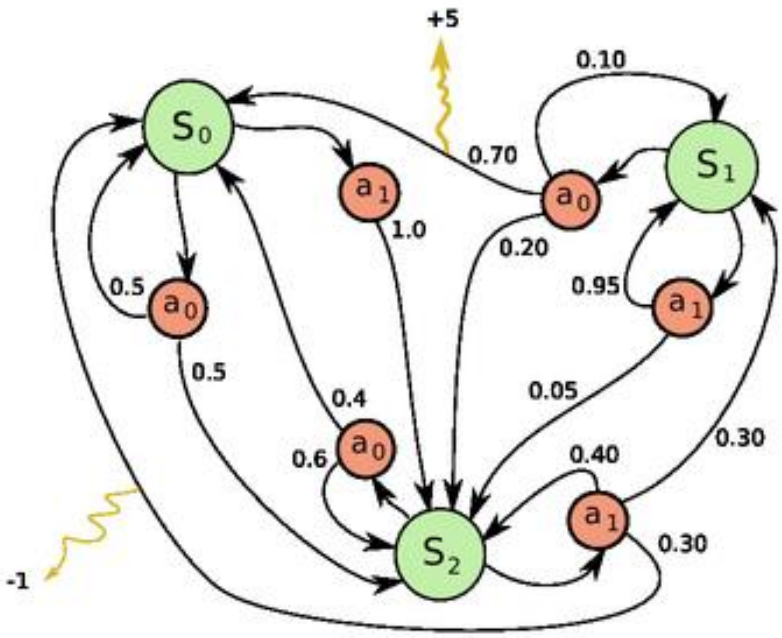


機器學習（ML）

- 目的：從現有的資料建構一個系統能夠做出最佳的決定
- 數學上而言，機器學習就是給定input X，想要去學一個函數f(X)能夠產生我們要的結果 Y
  - X: X光照片 ➔ Y：是否得癌症, classification
  - X: 金融新聞 ➔ Y：股票指數, regression
  - X: 很多人的照片 ➔ Y：把照片相似的分到同一群，clustering

規劃 (planning)

- 為了達成某個目的，人工智慧的代理人(Agent)根據環境做出相對應的行動
- Markov Decision Process



From Wikipedia

搜尋與最佳化

- 在許多可能性中，找出最好的選擇
  - 下棋：在所有可能的下一步中，搜尋出最有可能勝利的一步
  - 機器人足球：在所有可能的行動裡，選擇最有可能得分的行動
  - 益智問答：在所有可能的答案裡，選擇最有可能是正確
- 能夠被model成最佳化的問題，基本上AI就可以解
  - 關鍵在於搜尋與最佳化的速度

人工智慧與其他領域的結合

- 哲學（邏輯，認知，自我意識）
- 語言學（語法學，語意學，聲韻學）
- 統計與數學（機器學習）
- 控制（最佳化）
- 經濟學（Game Theory）
- 神經科學（類神經網路）
- 資訊工程（AI-driven CPU, 雲端計算）
- 資料科學（資料探勘）

應用一：推薦系統

- 推薦產品給客戶
  - E-commence sites
  - Service providers
- 推薦使用者給使用者
  - Match maker
  - Investment/donation
- 基本上，能夠對兩方做出智慧型的推薦
  - Recommending resume to job
  - Recommending course to students

應用2：物聯網（IoT）

- IoT 包括三個元素：資料，連結以及設備（感測器）
- 相關應用有
  - 智慧都市 (parking, traffic, waste management)
  - 災害防治 (abnormal PM2.5 level, electromagnetic filed levels, forest fire, earthquake, etc)
  - 健康 (monitoring, fall detection, etc)
  - 安全 (access control, law enforcement, terrorist identification)
  - 智慧農業

應用三：SOLOMO

- SOLOMO➔Social, Local, Mobile
- 許多相關應用
  - 智慧導遊
  - Nike+GPS: 路徑追蹤, 馬拉松運動轉播
  - 智慧廣告看板

應用4: 零售 & E-commence

- 智慧型產品規劃
- 智慧倉儲
- 貨品運輸最佳化
- 商品追蹤

Agenda

- 人工智慧的定義與歷史
- 人工智慧現在的發展與應用
- 人工智慧的未來：從機器學習到機器發明

AI的未來：從機器學習到機器發明

Moravec‘s Paradox

- 對人類而言很簡單的事情，對電腦而言可能很難；
  - 肢體與眼協調（如爬樓梯，踢足球）
  - 瞭解幽默與嘲諷
  - 人臉辨識
  - 對語意的深層瞭解
- 對人類很難的事情，對電腦而言可能很簡單。
  - 在大量資料中搜尋與分析
  - 下棋
  - 複雜的邏輯推論

停留在搜尋階段的人工智慧

- 在人工智慧網路交友的網站上，一個女生開出徵婚條件有兩點：1.要帥 2.要有車，輸入電腦去幫她搜尋，結果出現：
- 象棋!!
- 她看了不太滿意，於是改輸入：1.要高 2.要有房子，電腦跑一跑，結果出現：
- 台北101
- 她又改輸入：1.要MAN 2.要有安全感，電腦回她：
- 超人！！
- 她不死心，最後把以上所有條件都打進去：電腦回她
- 在101下象棋的超人!!

AI的未來兩大研究走向

- 達成更多超越人類的成就
  - 更精準的預測
  - 更正確的決策
- 達成原本對人類而言就很容易的任務
  - Computer vision
  - Robotics
  - Natural Language Understanding

From Learning to Discovery

- Learning：”The act, process, or experience of gaining knowledge”.
  - 學習是正常人在生活中都能體驗的經驗
- Discovery: “To obtain knowledge or awareness of something not known before.”
  - 發現或是發明是特殊人群在特殊時空下的產物如：愛迪生，牛頓，福爾摩斯。

Why Machine Discovery?

- Def: Exploit machines to perform or assist human beings to perform discovery
- Benefits:
  - High reward
  - Tolerates low precision and recall
- Challenge (why it is more difficult than ML?):
  - No standard solution available
  - No labeled data available

Two Types of Discovery

- Machine as a scientist

- Machine as a detective

Machine as Scientists

As mathmatician: Graph Theory（GRAFFITI，1990 - , Fajtlowicz）

- Made conjectures in Graph Theory such as
  chromatic_number(G) + radius(G) =< max_degree(G) + frequency_of_max_degree(G)
- Discovered > 800 conjectures, which have led to 60 math papers and 1 thesis publications to proof or disproof them

As physicists: ASTRA (1998 - , Kocabas and Langley)

- suggested novel and interesting fusion reactions, and generated reaction pathways for helium, carbon, and oxygen that do not appear in the scientific literature

As astronomers: SKICAT (1995 - , Kennefick)

- aiming at detecting quasars in digital sky surveys, and has found 5 new quasars.

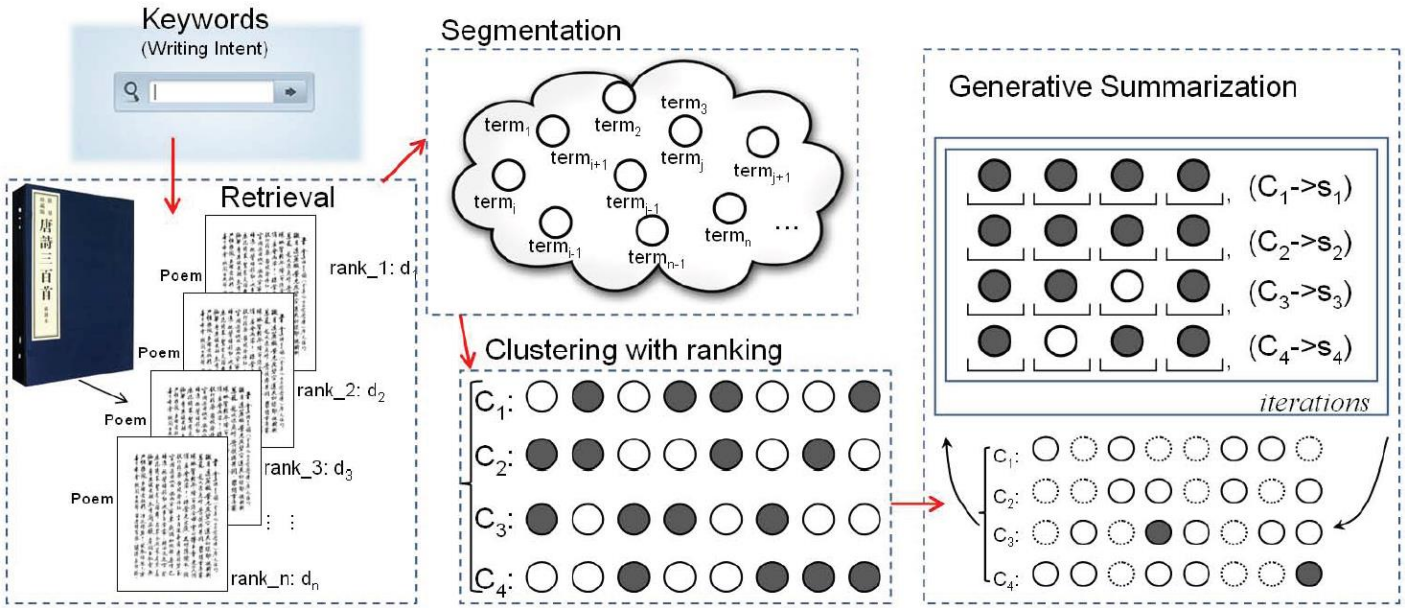
As Linguistics：魯汶古文的線性讀法

魯汶古文是西元前1300-700在敘利亞北方使用的語言

它是二維文字，還沒有完全解碼

physicists

As Poet: 電腦創作中國古典詩



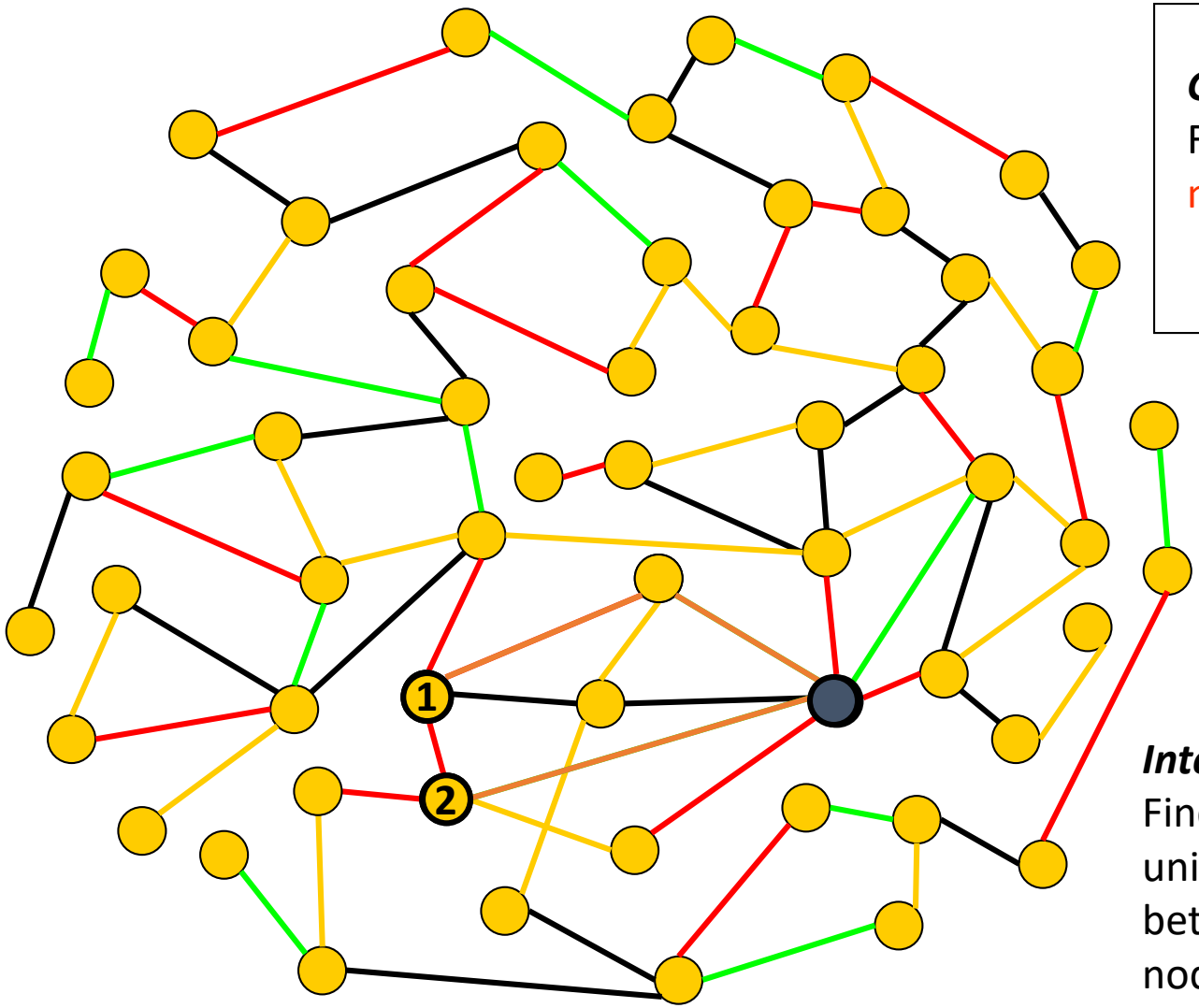
故人	思鄉	回文詩
千里行路難	何處不自由	行人路道路人行
平生不可攀	故土木蘭舟	別有心人心有別
相逢無人見	落日一回首	寒風幾度幾風寒
與君三十年	孤城不可留	

Two Types of Discovery

- Machine as scientists

- Machine as detectives

Abnormal Instance Discovery From Social Networks



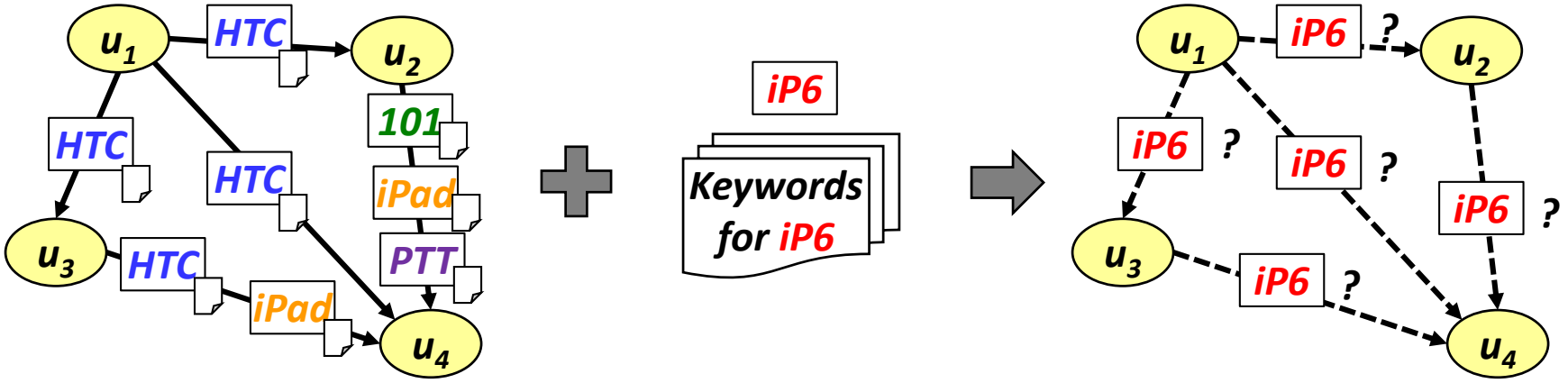
Global global node discovery: Finding the most abnormal node(s) in the network

Local abnormal node discovery: Finding the most abnormal node(s) connected to a given node

Interesting feature discovery: Finding the most interesting or unique features (e.g. paths) between two nodes or w.r.t a node

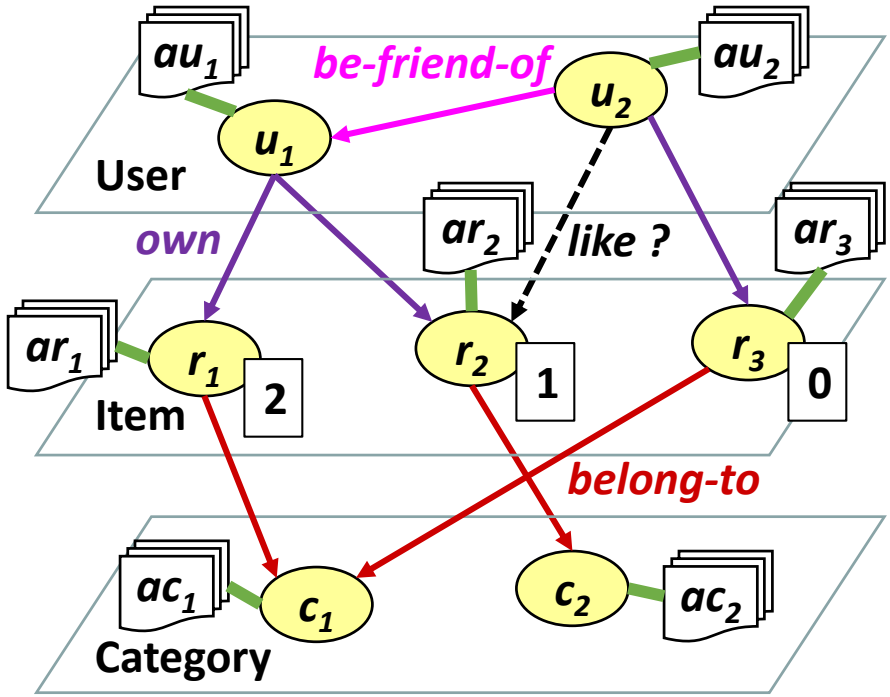
Novel Topic Diffusion Discovery (ACL 2012)

- Predicting diffusions for novel topic
  - Predict diffusions of "*iPhone 6*" before any such diffusion occurs using
    - Links of existing topics with contents
    - Keywords of novel topics
- Intuition of our approach
  - We do have the diffusion data of other topics
  - Finding the connection between existing topics and novel topics using a LDA-based approach
  - Exploiting such connection for link prediction



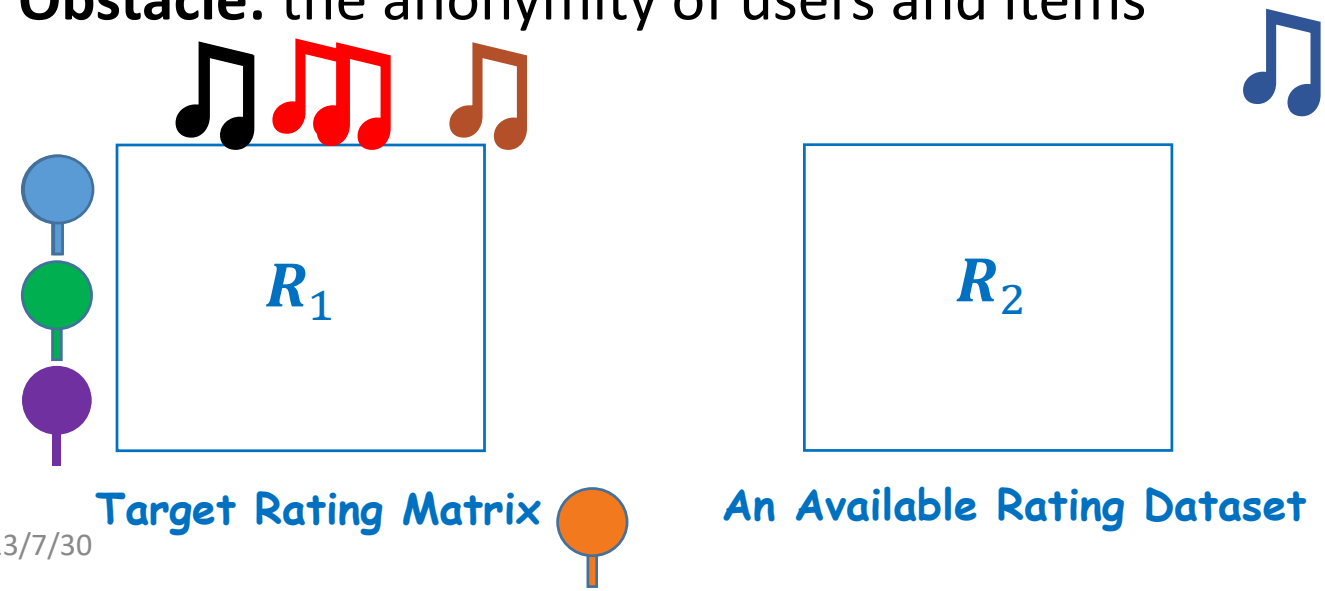
Unseen Links Discovery (KDD 2013)

- Individual opinion (ex. customer's preference) is valuable
  - sometimes concealed due to privacy (ex. Foursquare "*like*")
  - Fortunately, aggregative statistics (total count) is usually available
- Predict unlabeled relationship (unseen-type link) using
  - Heterogeneous social network
  - Attributes of nodes
  - Aggregative statistics

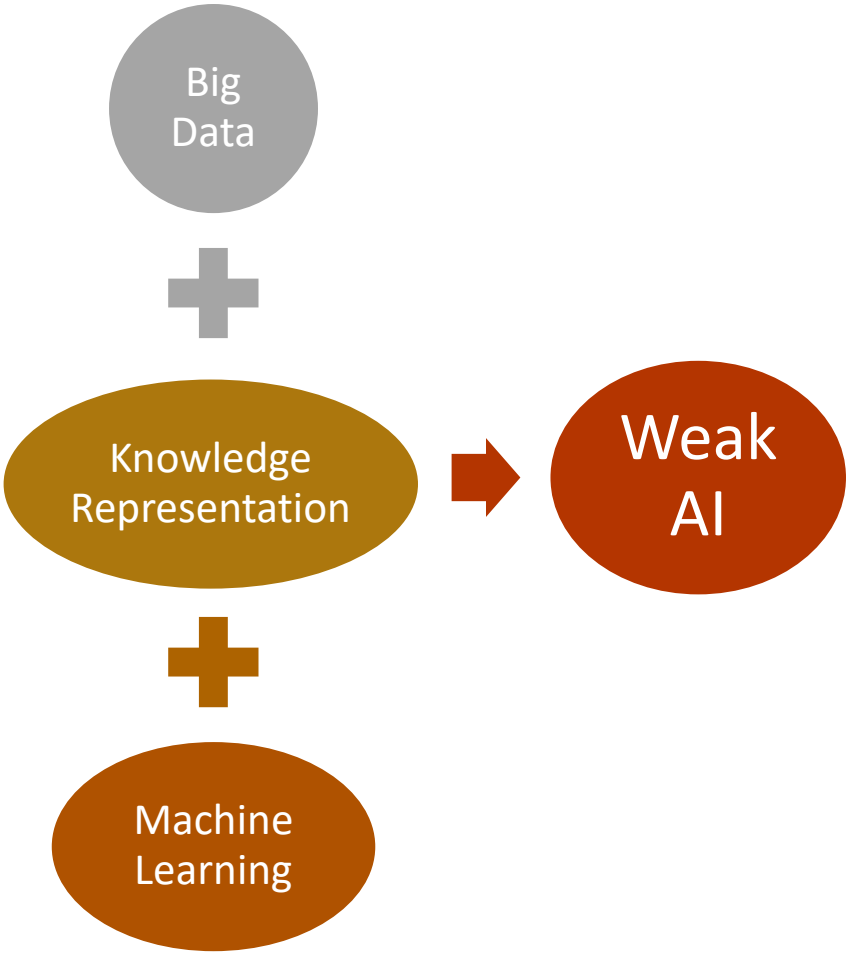


Matching Users and Items for Transfer Learning in Collaborative Filtering (KDD2014)

- **Assumptions:**
  - Two rating matrices
  - Modeling the same kind of preference
  - Certain portion of overlap in users and items
  - E.g. Using Blockbuster's record to enhance the performance of a NetFlix System
- **Obstacle:** the anonymity of users and items



Final Remark: 弱人工智慧的時代已經降臨



我對人工智慧帶來的未來有三個大膽的預測

2006: "I bet in 10 years we'll cure cancer, inhabit the moon, have world peace"

2016:

Halle Tecco

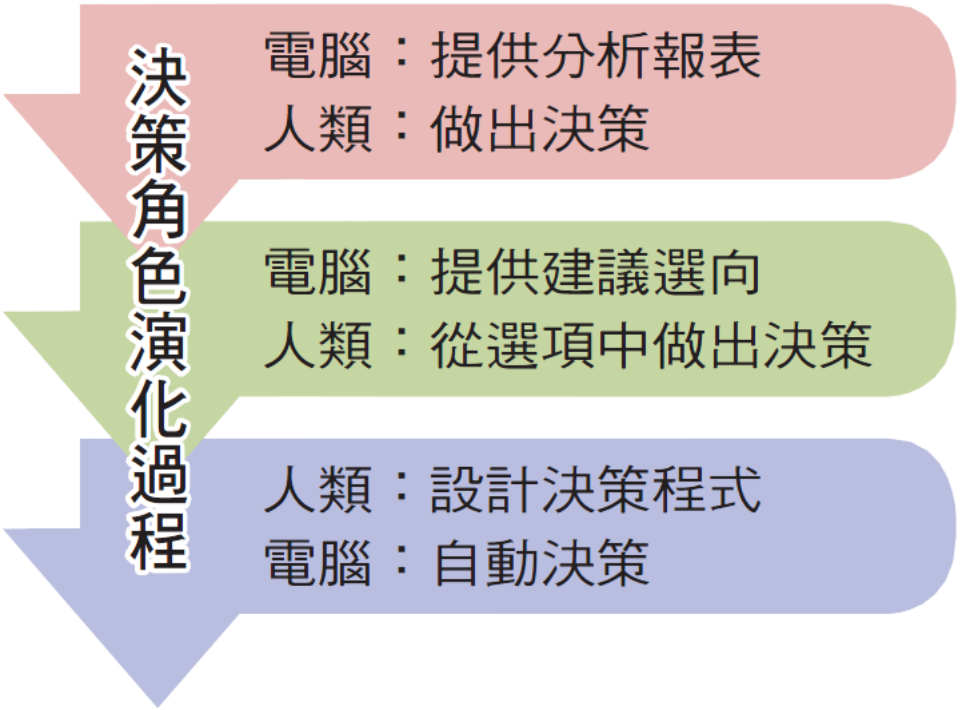
人工智慧革命

Machine will take over some non-trivial jobs that are used to be conducted by human beings

- Some trivial jobs will be safe ☺

決策角色的翻轉

- The decision makers will gradually shift from humans to machines



電腦在決策過程中角色的改變 °科學人2015

從機器學習到機器發明

- 人類智慧的演進 ：

  Machine memorizing

  Machine learning

  Machine discovering

- 不久的將來，許多科學新知將會由AI發現！

你所不知道的機器學習

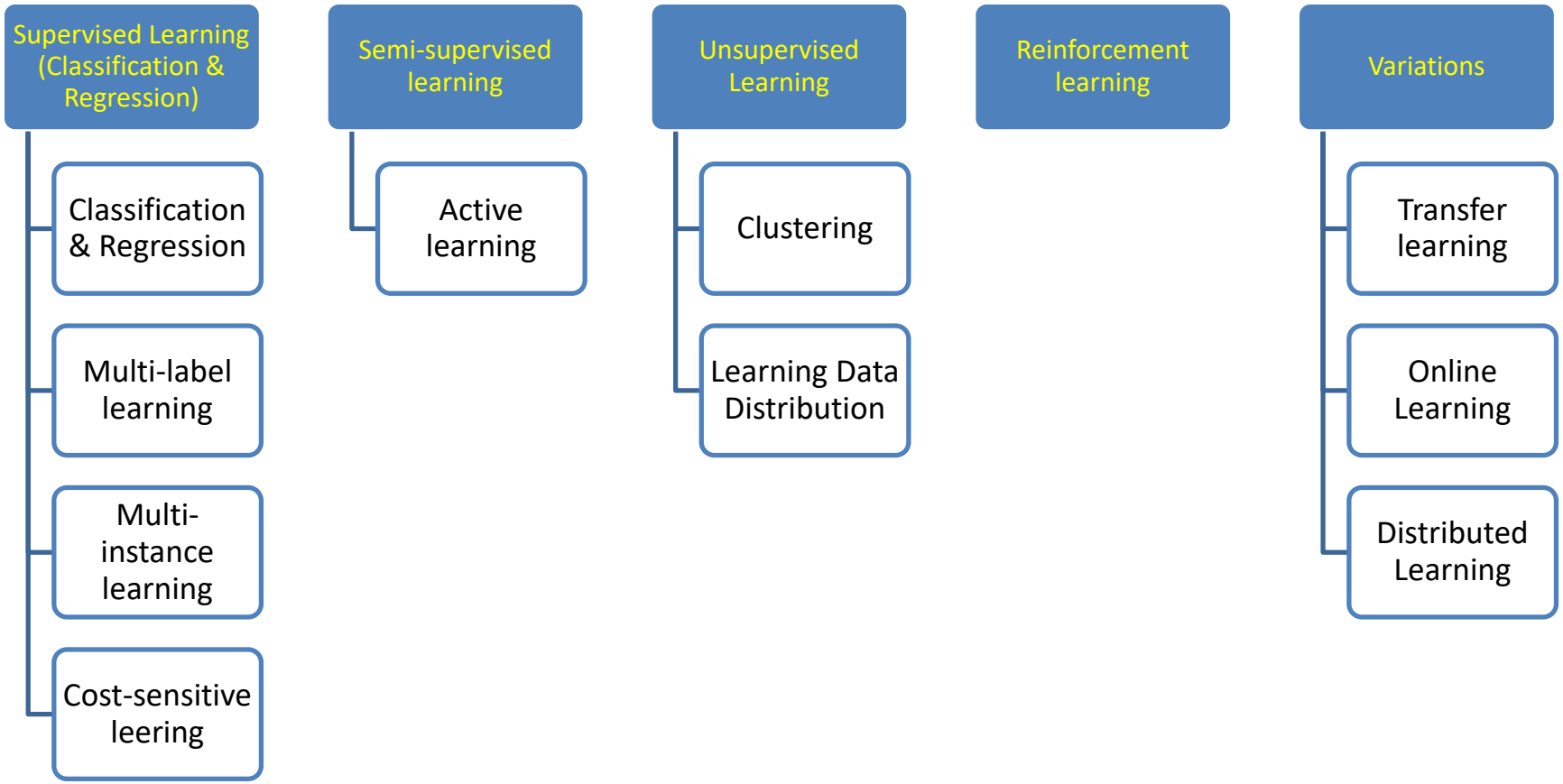
Shou-De Lin（林守德）

CSIE/GINM, NTU
sdlin@csie.ntu.edu.tw

What is Machine Learning

- ML tries to optimize a performance criterion using example data or past experience.
- Mathematically speaking: given some data X, we want to learn a function mapping f(X) for certain purpose
  - f(x)= a label y → classification
  - f(x)= a set Y in X → clustering
  - f(x)=p(x) → probabilistic graphical model
  - f(x)= a set of y → multi-label classification
  - …
- ML techniques tell us how to produce high quality f(x), given certain objective and evaluation metrics

A variety of ML Scenarios



Supervised Learning

- Given: a set of <input, output> pairs
- Goal: given an unseen input, predict the corresponding output
- For example:
  1. Input: X-ray photo of chests, output: whether it is cancerous
  2. Input: a sentence, output: whether a sentence is grammatical
  3. Input: some indicators of a company, output: whether it will make profit next year
- There are two kinds of outputs an ML system generates
  - Categorical***: classification problem (E1 and E2)***
    - *Ordinal outputs: small, medium, large*
    - *Non-ordinal outputs: blue, green, orange*
  - Real values***: regression problem (E3)***

Classification (1/2)

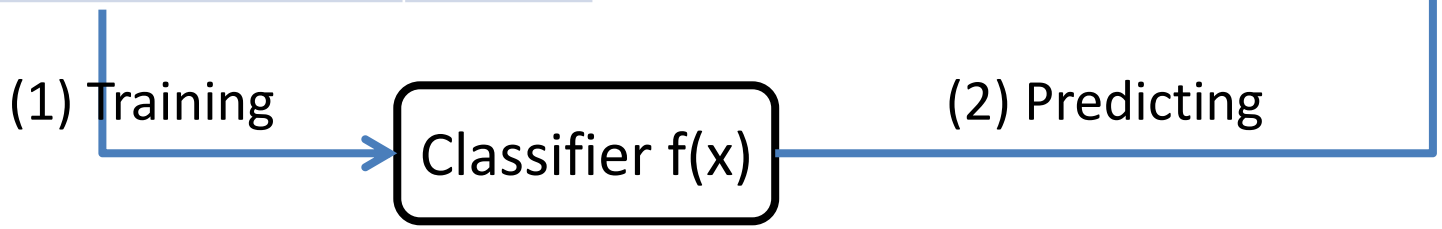
- It's a supervised learning task that, given a real-valued feature vector x, predicts which class in $\mathbf{C}$ may be associated with x.
- $|\mathbf{C}|=2$ ➔ Binary Classification
  $|\mathbf{C}|>2$ ➔ Multi-class Classification
- Training and predicting of a binary classification problem:

Training set (Binary Classification)

Feature Vector ($x_i \in \mathbf{R}^d$)	Class
x_1	+1
x_2	-1
…	…
x_{n-1}	-1
x_n	+1

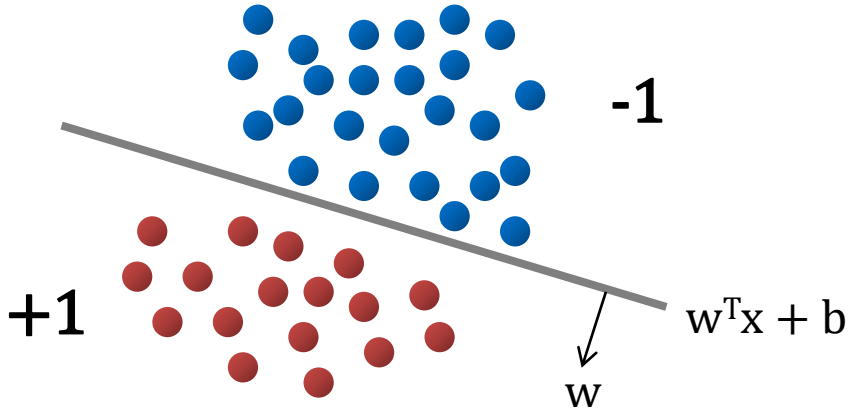
A new instance

Feature Vector ($x_{new} \in \mathbf{R}^d$)	Class
x_{new}	?



Classification (2/2)

- A classifier can be either linear or non-linear
- The geometric view of a linear classifier

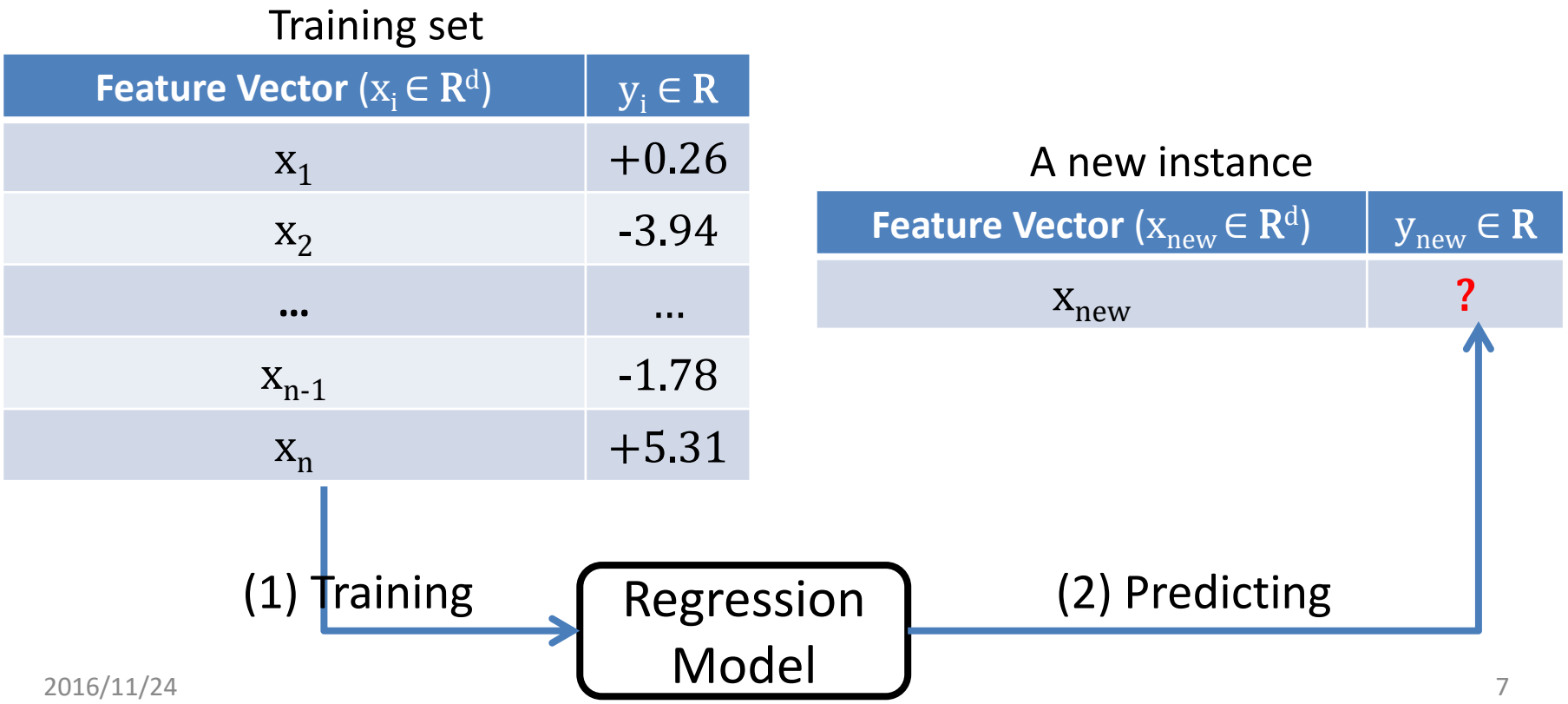


We will talk more about classification later !!

- Famous classification models:
  - k-nearest neighbor (kNN)
  - Decision Tree (DT)
  - Support Vector Machine (SVM)
  - Neural Network (NN)
  - …

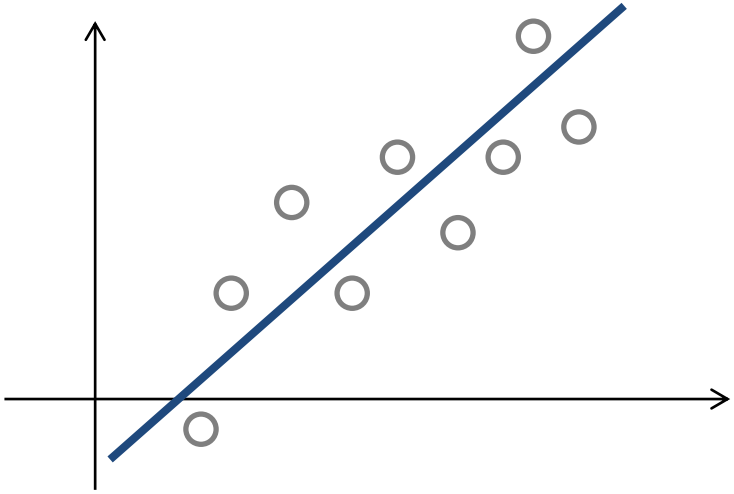
Regression (1/2)

- A supervised learning task that, given a real-valued feature vector x, predicts the target value $y \in R$.
- Training and predicting of a regression problem:



Regression (2/2)

- The geometric view of a linear regression function



- Some types of regression: linear regression, support vector regression, …

Multi-label Learning

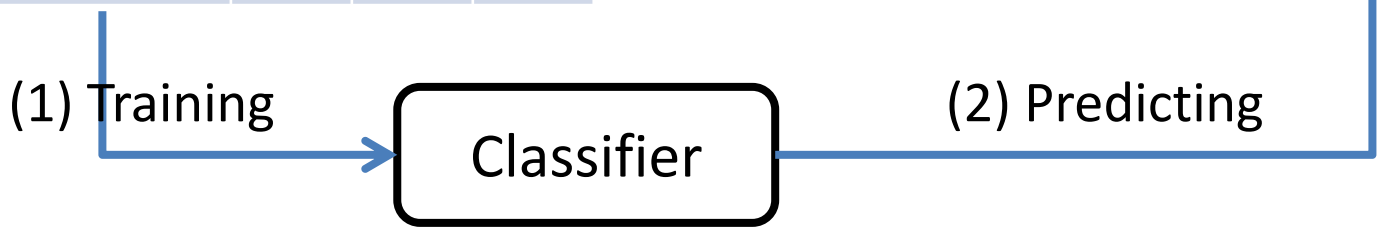
- A classification task in that an instance is associated with a set of labels, instead of a single label.

Training set

Feature Vector ($x_i \in R^d$)	ℓ_1	ℓ_2	ℓ_3
x_1	+1	-1	+1
x_2	-1	+1	-1
…	…	…	…
x_{n-1}	+1	-1	-1
x_n	-1	+1	+1

A new instance

Feature Vector ($x_{new} \in R^d$)	ℓ_1	ℓ_2	ℓ_3
x_{new}	?	?	?



- Existing models: Binary Relevance, Label Powerset, ML-KNN, IBLR, …

Multimedia tagging

- Many websites allow the user community to add tags, thus enabling easier retrieval.

Example of a tag cloud: the beach boys, from Last.FM (Ma et al., 2010)

Cost-sensitive Learning

- A classification task with non-uniform cost for different types of classification error.
- Goal: To predict the class C* that minimizes the expected cost rather than the misclassification rate

$$C^* = \arg\min_j \sum_k P(Y = C_k \mid x) L_{jk}$$

- An example cost matrix L : medical diagnosis

L_{jk}	Actual Cancer	Actual Normal
Predict Cancer	0	1
Predict Normal	10000	0

- Methods: cost-sensitive SVM, cost-sensitive sampling

Examples for Cost-sensitive Learning

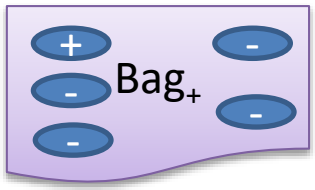
- Highly non-uniform misclassification costs are very common in a variety of challenging real-world machine learning problems
  - Fraud detection
  - Medical diagnosis
  - Various problems in business decision-making.

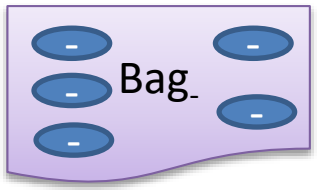
Credit cards are one of the most famous targets of fraud. The cost of missing a target (fraud) is much higher than that of a false-positive.

Hung-Yi Lo, Shou-De Lin, and Hsin-Min Wang, "Generalized k-Labelsets Ensemble for Multi-Label and Cost-Sensitive Classification," IEEE Transactions on Knowledge and Data Engineering.

Multi-instance Learning (1/2)

- A supervised learning task in that the training set consists of *bags of instances*, and instead of associating labels on instances, labels are *only* assigned to bags.
- In the binary case,





Positive bag → at least one instance in the bag is positive

Negative bag → all instances in the bag are negative

- The goal is to learn a model and predict the label of a new instance or a new bag of instances.

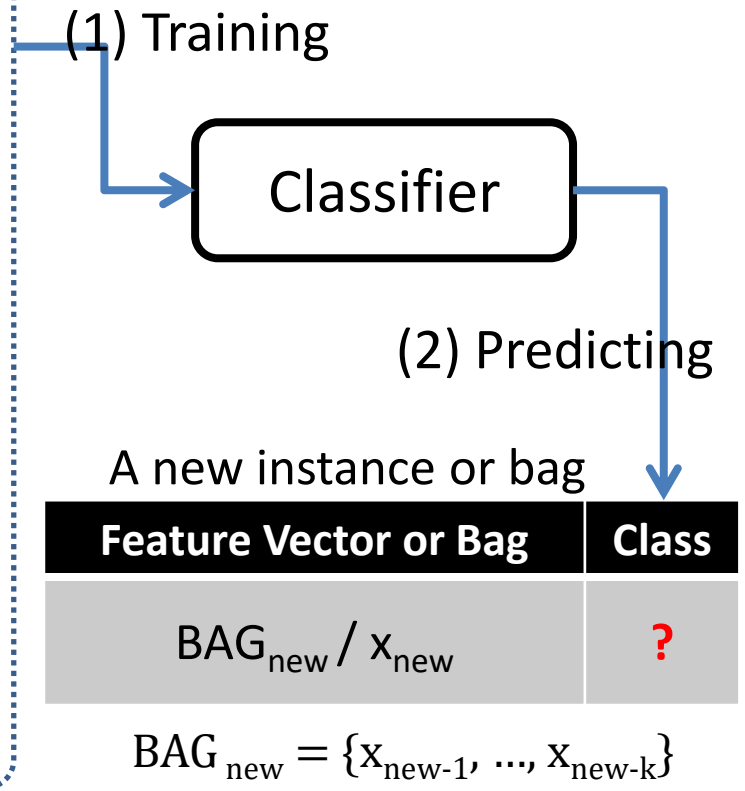
Multi-instance Learning (2/2)

- Training and Prediction

Training Set

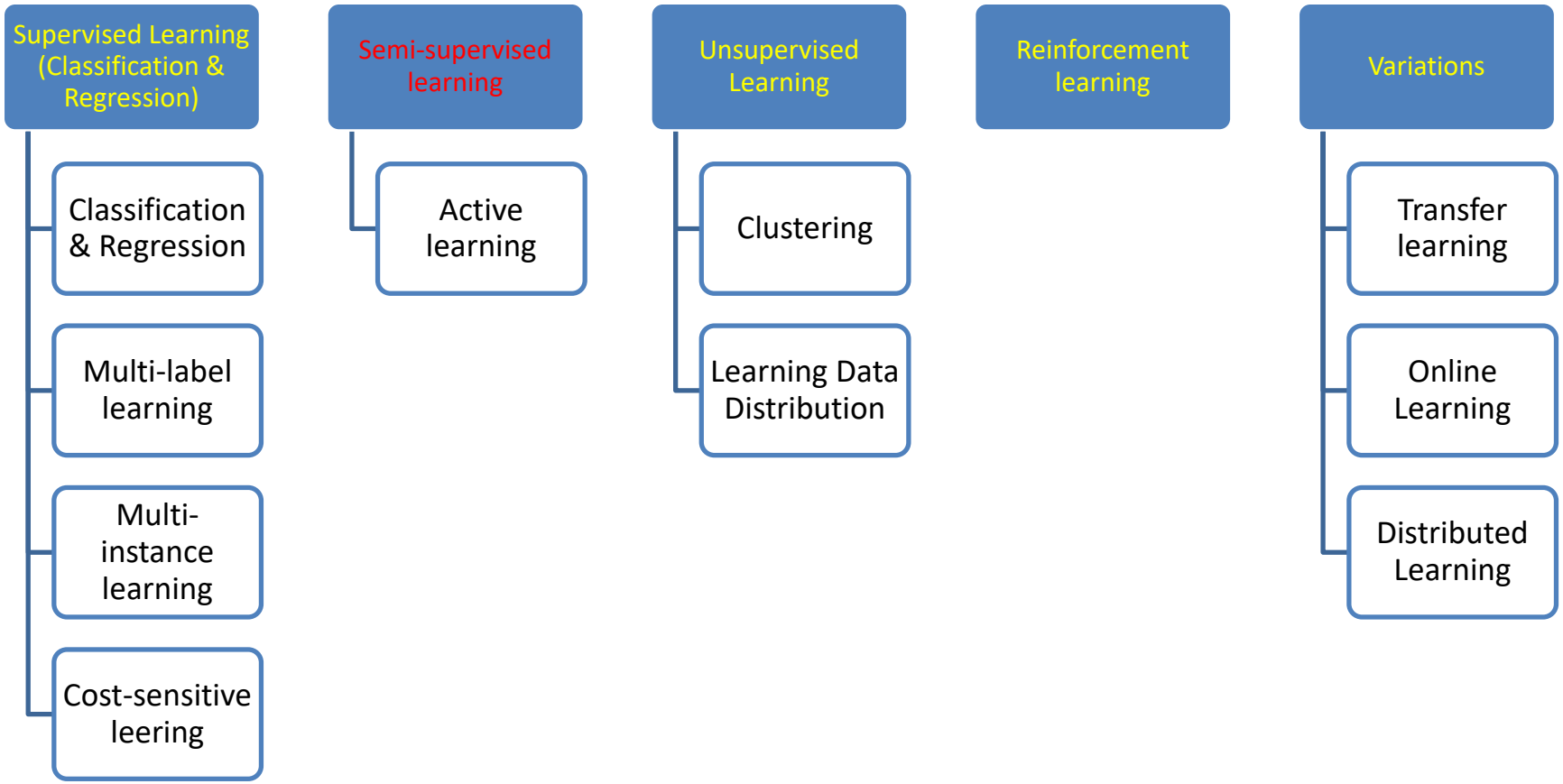
Feature Vector ($x_i \in R^d$)	Bag
x_1	1
x_2	1
x_3	2
x_4	2
x_5	2
…	…
x_{n-1}	m
x_n	m

Bag	Class
1	+1
2	-1
…	…
m	-1



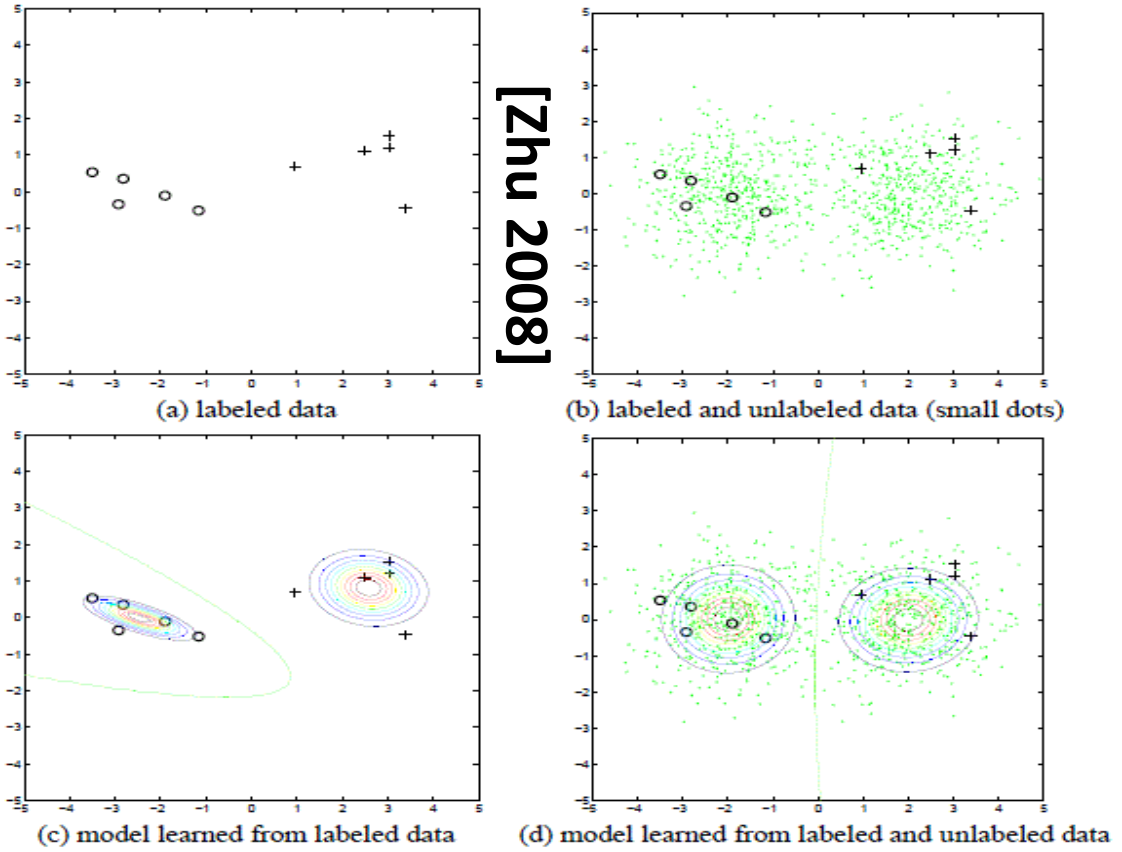
- Some methods: Learning Axis-Parallel Concepts, Citation kNN, mi-SVM, MI-SVM, Multiple-decision tree, …

A variety of ML Scenarios



Semi-supervised Learning

- Only a small portion of data are annotated (usually due to high annotation cost)
- Leverage the unlabeled data for better performance



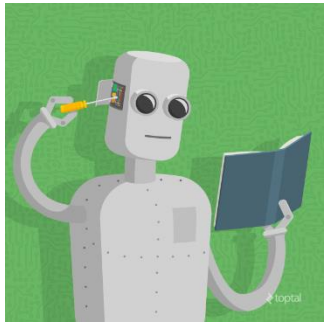
Active Learning

- Achieves better learning with fewer labeled training data via actively selecting a subset of unlabeled data to be annotated

Unlabeled Data

Expert / Oracle

Learning Algorithm



Request for the Label of an Example

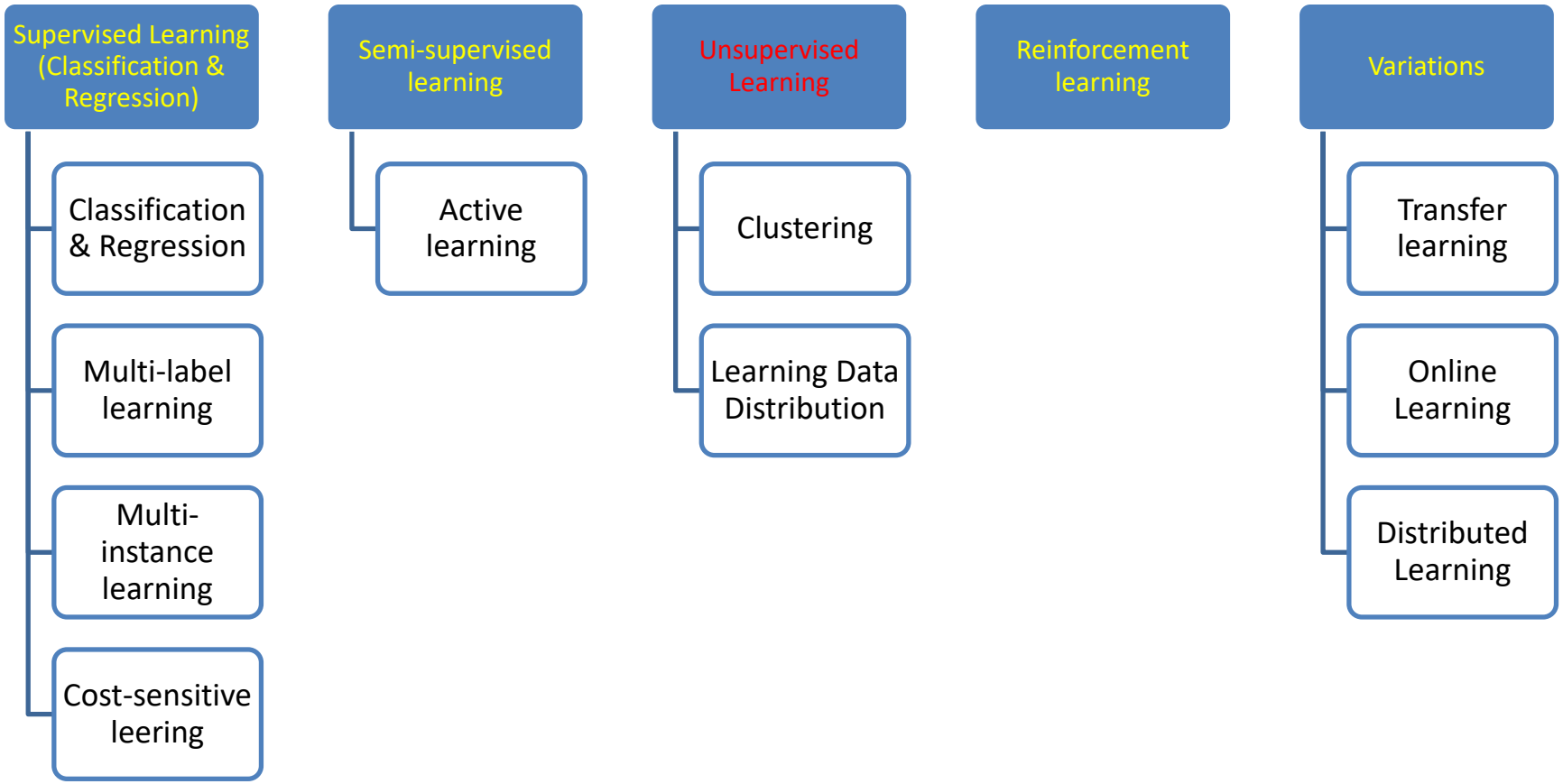
A Label for that Example

Request for the Label of another Example

A Label for that Example

Algorithm outputs a classifier

A variety of ML Scenarios



Unsupervised Learning

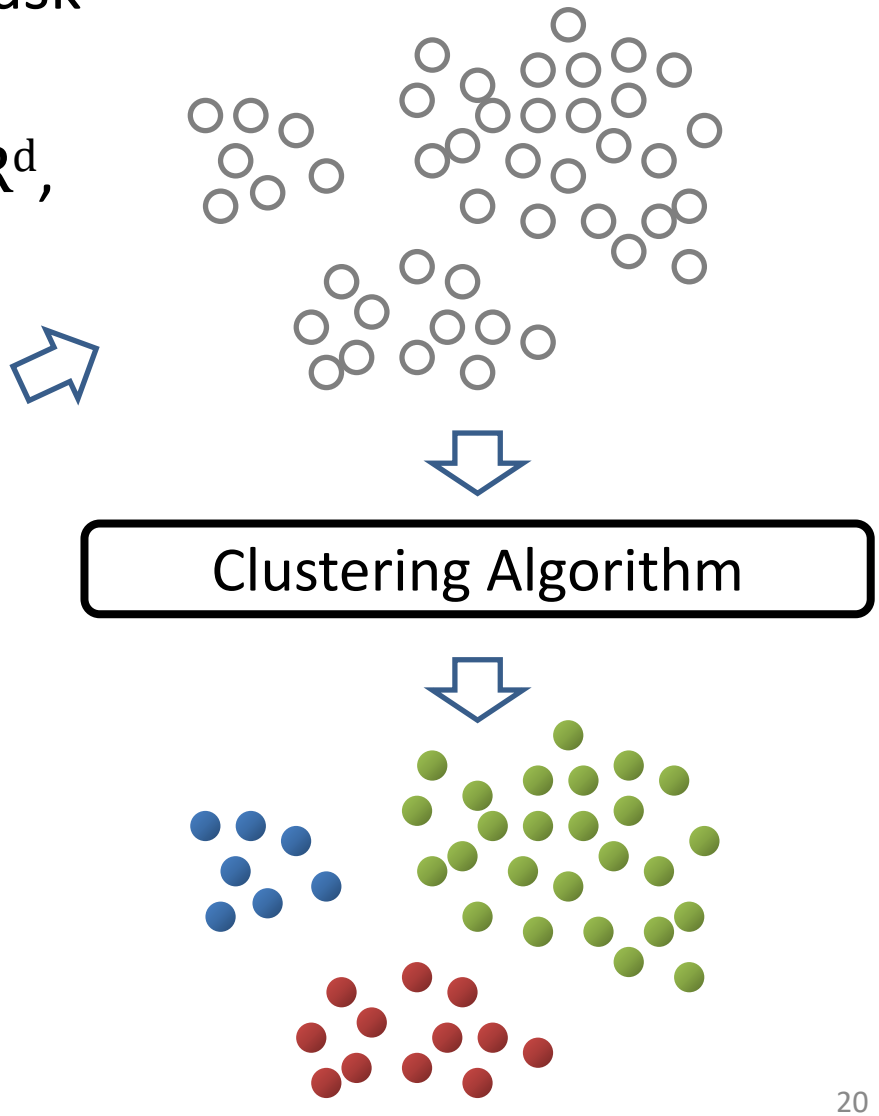
- Learning without teachers (presumably harder than supervised learning)
  - Learning “what normally happens”
  - Think of how babies learn their first language (unsupervised) comparing with how people learn their 2nd language (supervised).
- Given: a bunch of input X (there is no output Y)
- Goal: depending on the tasks, for example
  - Estimate P(X) ➔ then we can find augmax P(X) ➔PGM
  - Finding $P(X_2|X_1)$➔ we can know the dependency between inputs ➔ PGM
  - Finding $Sim(X_1,X_2)$ ➔ then we can group similar X’s ➔ clustering

Clustering

- An unsupervised learning task
- Given a finite set of real-valued feature vector $S \subset R^d$, discover clusters in S

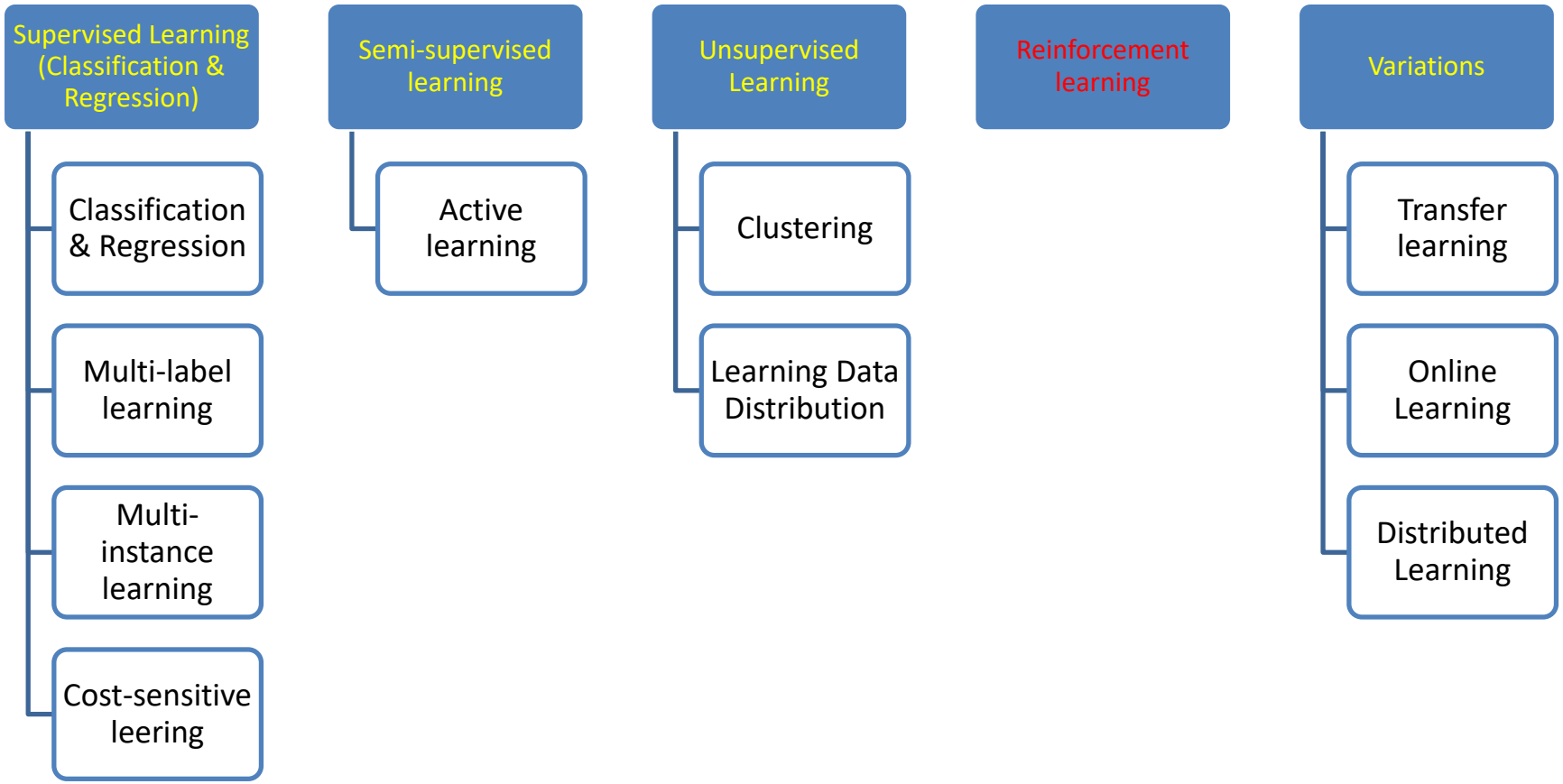
S

Feature Vector ($x_i \in R^d$)
x_1
x_2
…
x_{n-1}
x_n



- K-Means, EM, Hierarchical classification, etc

A variety of ML Scenarios



Reinforcement Learning (RL)

- RL is a “decision making” process.
  - How an agent should make decision to maximize the long-term rewards
- RL is associated with a sequence of states X and actions Y (i.e. think about Markov Decision Process) with certain “rewards”.
- It’s goal is to find an optimal policy to guide the decision.

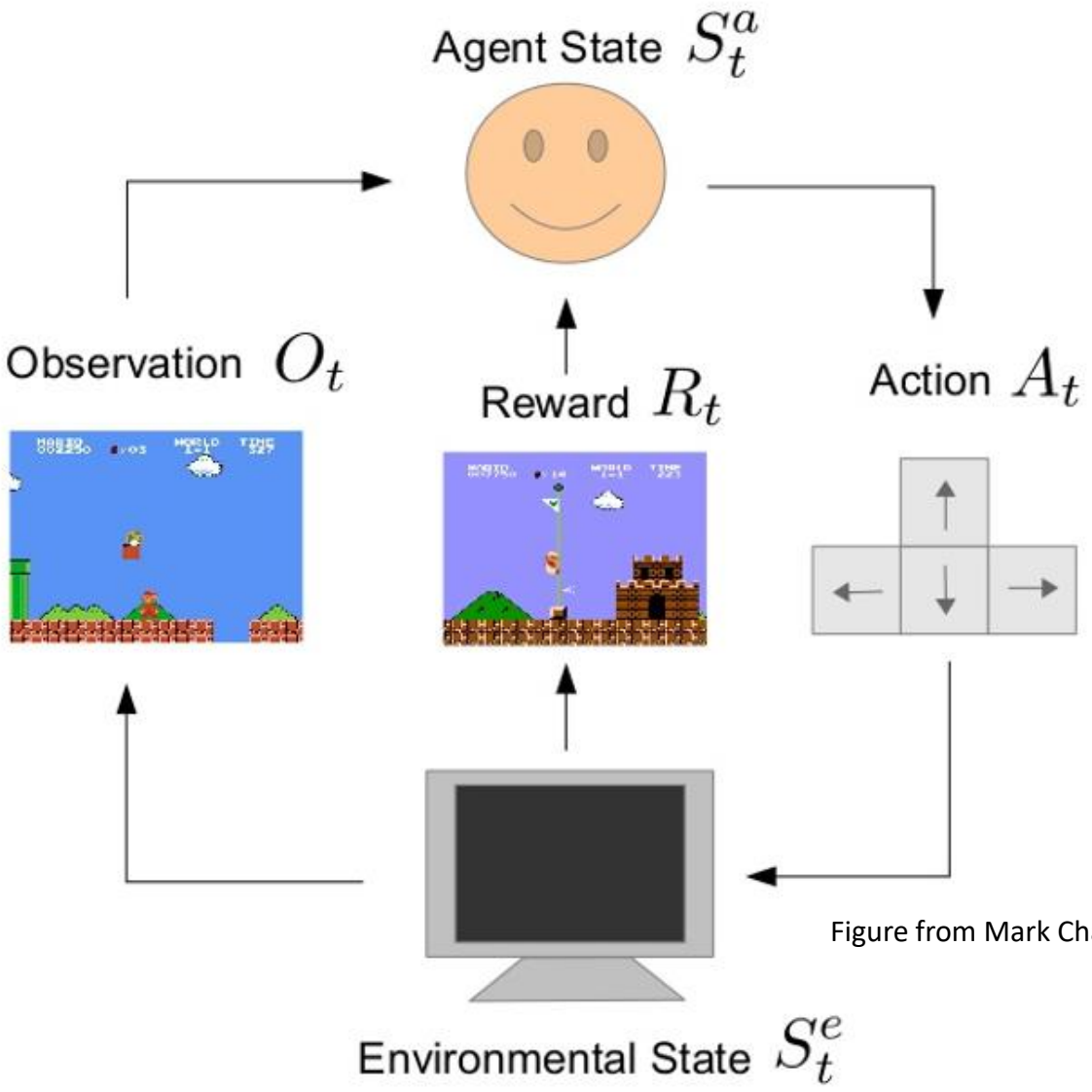
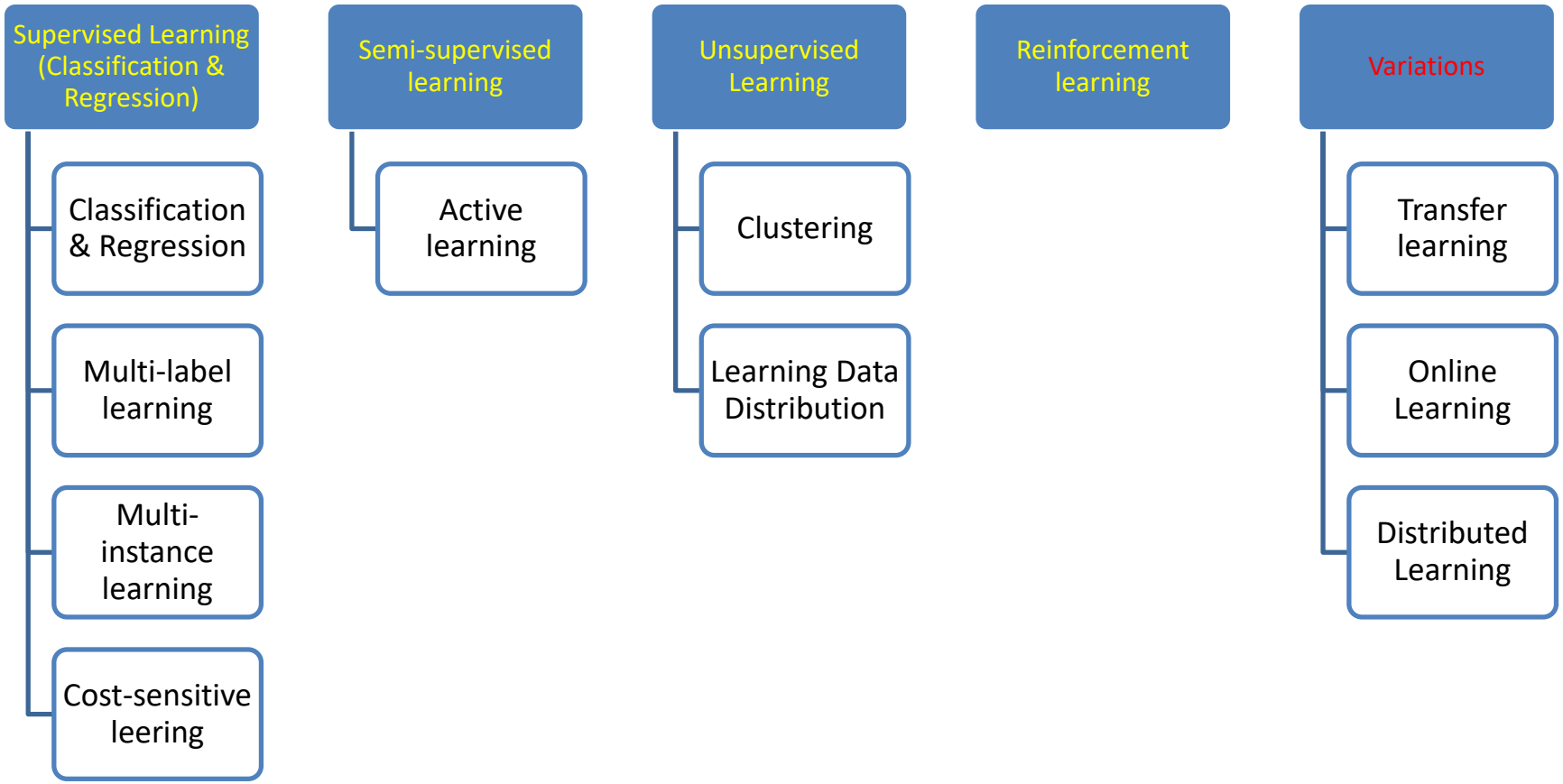


Figure from Mark Chang

AlphaGo: SL+RL

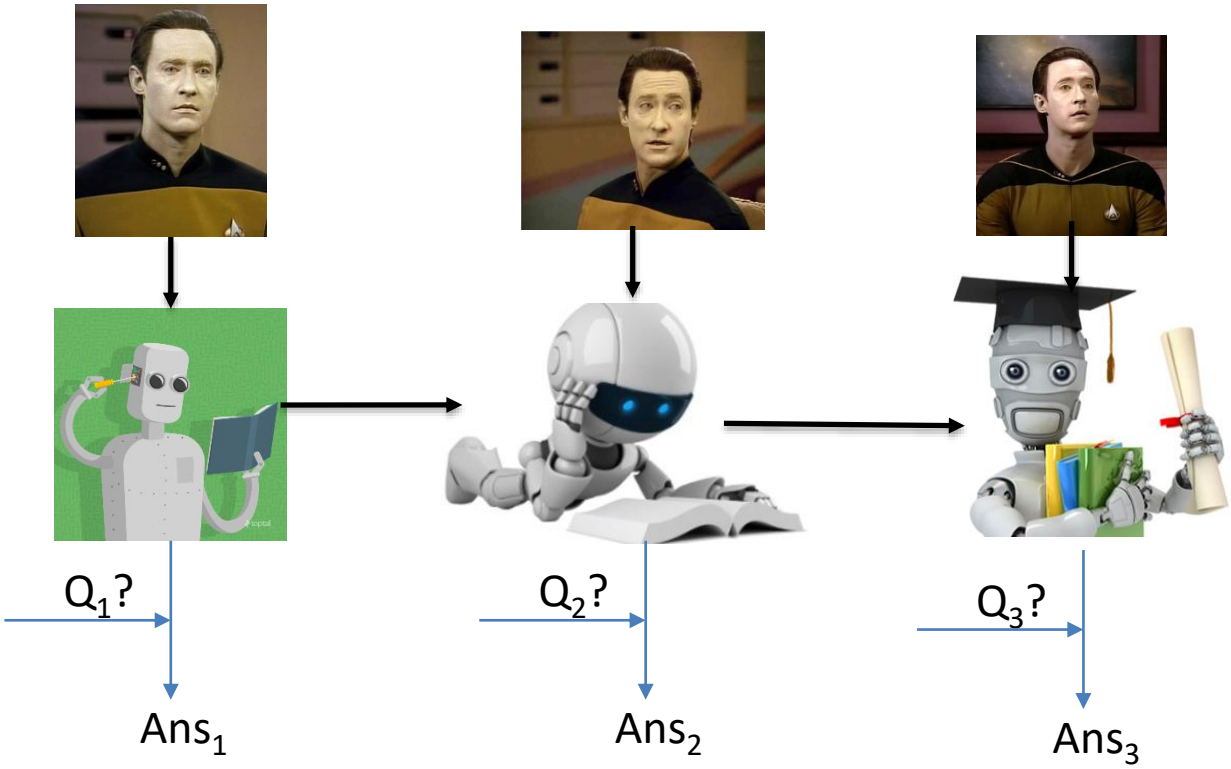
- 1st Stage:天下棋手為我師 (Supervised Learning)
  - Data: 過去棋譜
  - Learning: f(X)=Y, X: 盤面, Y:next move
  - Results: AI can play Go now, but not an expert
- 2nd Stage:超越過去的自己(Reinforcement Learning)
  - Data: generating from playing with 1st Stage AI
  - Learning: Observation→盤面, reward→if win, action→ next move

A variety of ML Scenarios

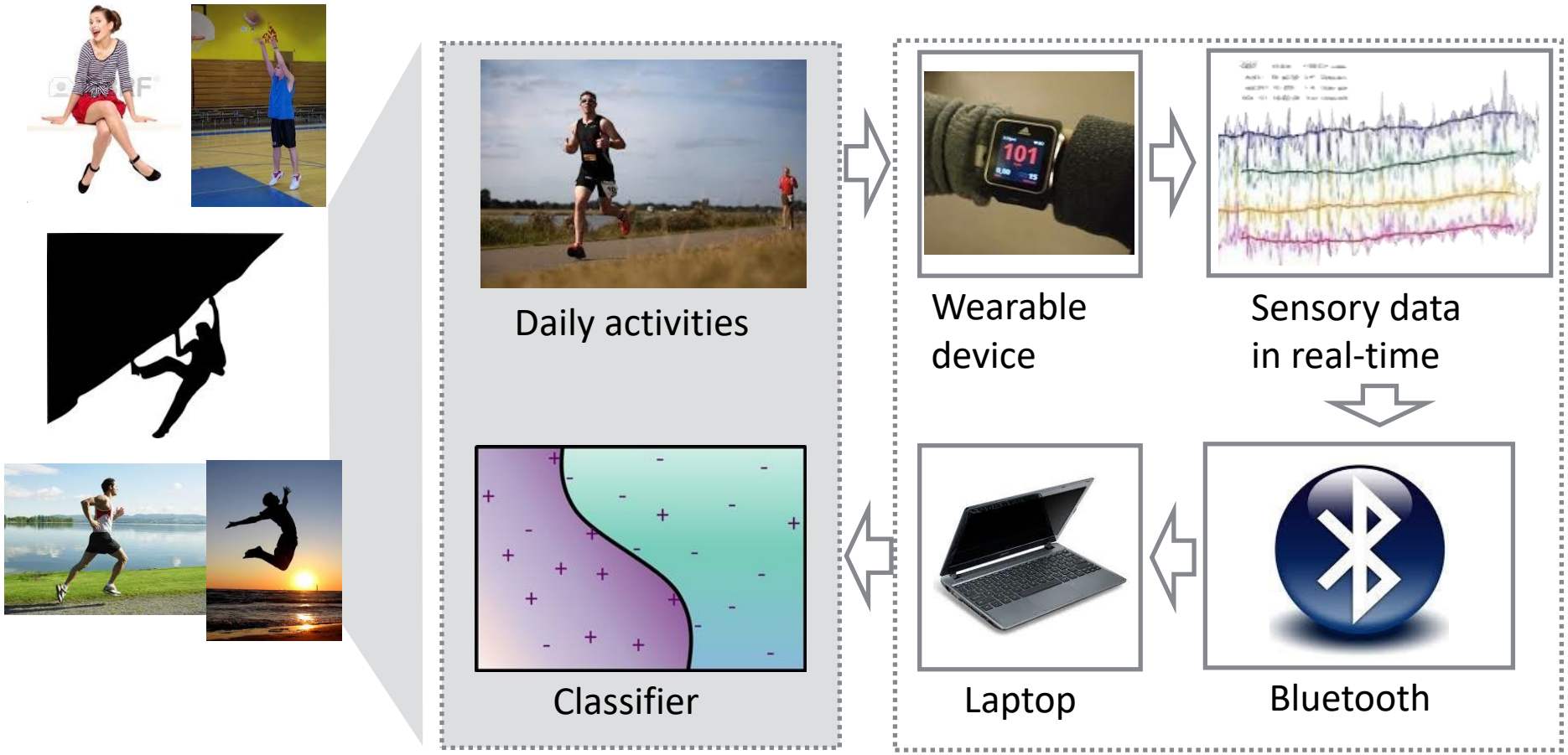


Online Learning

- Data arrives incrementally (one-at-a-time)
  - Once a data point has been observed, it might never be seen again.
  - Learner makes a prediction on each observation.
- Time and memory usage cannot scale with data.
  - Algorithms may not store previously seen data and perform batch learning.
  - Models resource-constrained learning, e.g. on small devices.



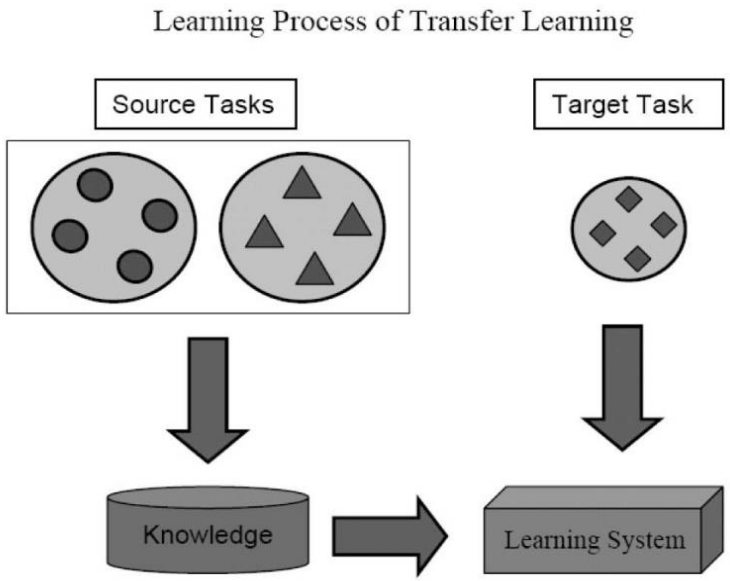
Scenario: Human activity recognition using data from wearable devices (one of the dataset we have experimented)



- There are a variety of models to be learned (individual * activity)
- Data are coming incrementally while we don't want to transmit everything to server
- The incoming data are unlabeled

Transfer Learning

- Improving a learning task via incorporating knowledge from learning tasks in other domains with different feature space and data distribution.
- Reduces expensive data-labeling efforts
- Example: the knowledge for recognizing an airplane may be helpful for recognizing a bird.
- Approach categories:
  (1) Instance-transfer
  (2) Feature-representation-transfer
  (3) Parameter-transfer
  (4) Relational-knowledge-transfer

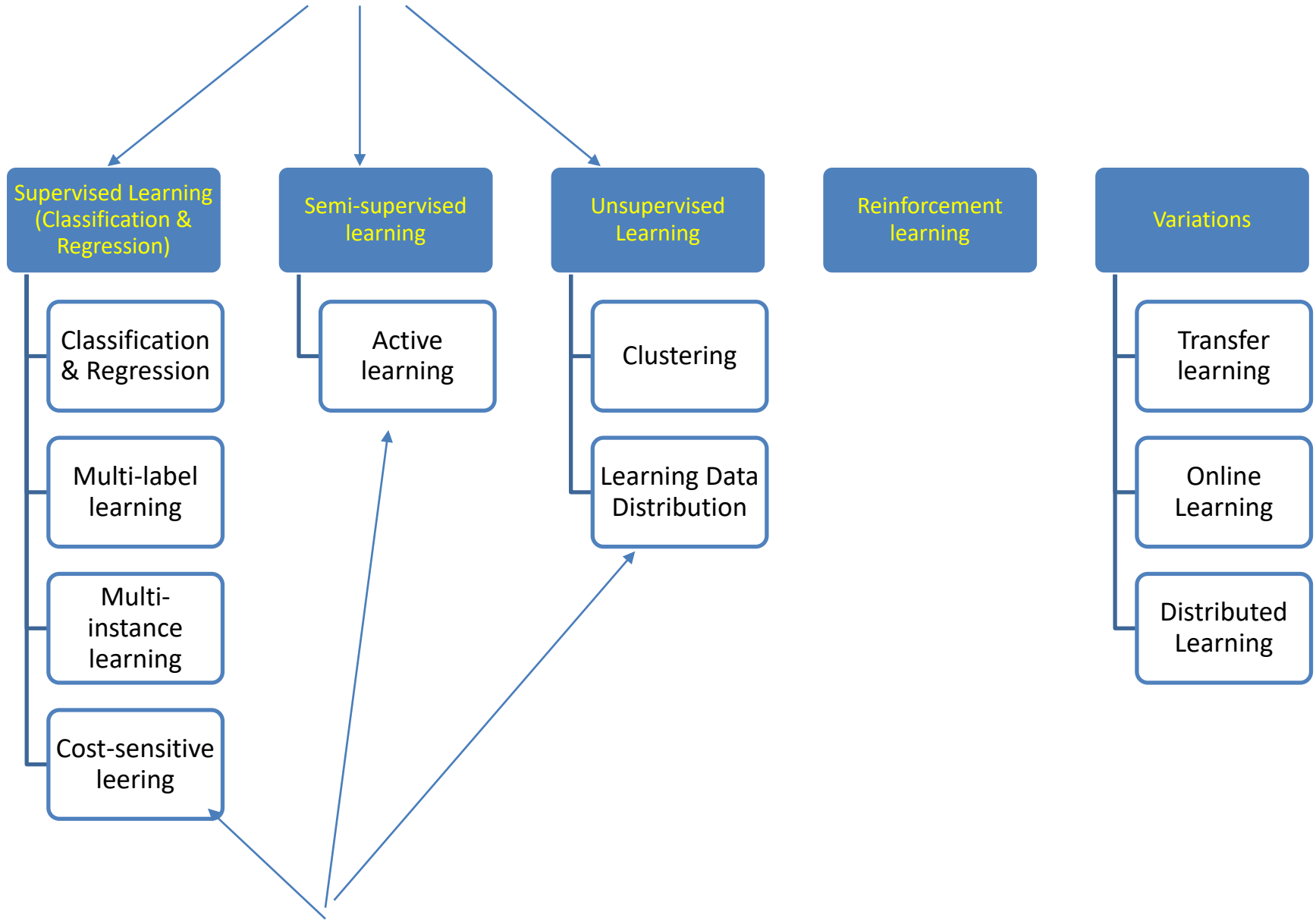


[Pan & Yang 2010]

Distributed Learning

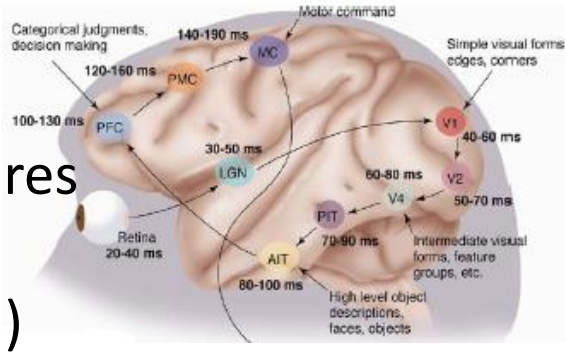
- Perform machine learning on multiple machines

Computation	Traditional Parallel Computing	Distributed Learning
# of machines	Few (10~100), communication cost can be ignored	Many (>1000), communication cost can be the bottleneck
Computational power	Powerful (cluster), dedicated	Ordinal (mobile), cannot be dedicated
Memory	Large	Small
Management	Strong control	Weak control

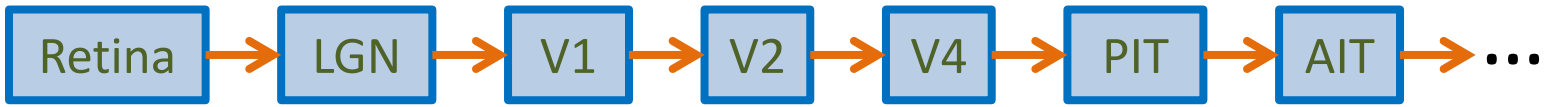

Deep Learning (learning representation of data)
Supervised Learning (Classification & Regression)
Semi-supervised learning
Unsupervised Learning
Reinforcement learning
Variations
Classification & Regression
Multi-label learning
Multi-instance learning
Cost-sensitive leering
Active learning
Clustering
Learning Data Distribution
Transfer learning
Online Learning
Distributed Learning
Probabilistic Graphical Model

Deep Learning

- Human brains perform much better than computers at recognizing natural patterns
- The brains learn to extract many layers of features
- Features in one layer are combined to form higher-level features in the layer above (Hinton)

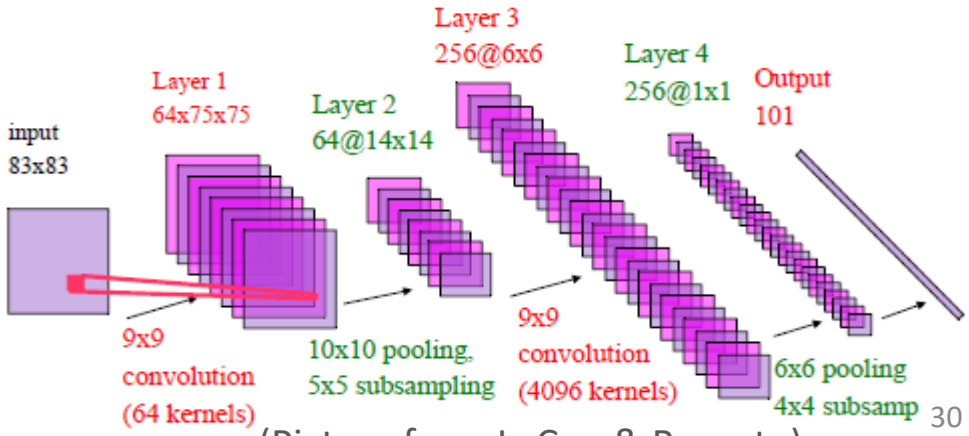


- Multiple processing stages in the visual cortex (LeCun & Ranzato):



- Inspired by human brains, *deep learning* aims to learn multiple layers of representation from data, with increasing level of abstraction

Convolutional Network



(Picture from LeCun & Ranzato)

如何訓練出好的機器學習模型？

Tips for Training a Good ML Model

- Feature Engineering
- Blending and Ensemble
- Training and Optimization
- Validation
- Novel ideas
- Smart and motivated personnel with significant amount of time investigated

Feature Engineering

- Feature engineering turns out to be one of the key strategy to improve the performance.
- The goal is to explicitly reveal important information to the model
  - domain knowledge might or might not be useful
- Original features → different encoding of the features → combined features
- Case study: KDD Cup 2010

Feature Preprocessing

- Categorical: expanded to binary features
  - Student, unit, problem, step, etc
  - E.g. 3310 students ➔ feature vector contains 3310 binary features
- Numerical: scaling (e.g. normalization to N(0,1), log (1+x), linear scaling, etc)
  - Check the original distribution before deciding how to perform scaling
  - Plot each features with different colors for different labels

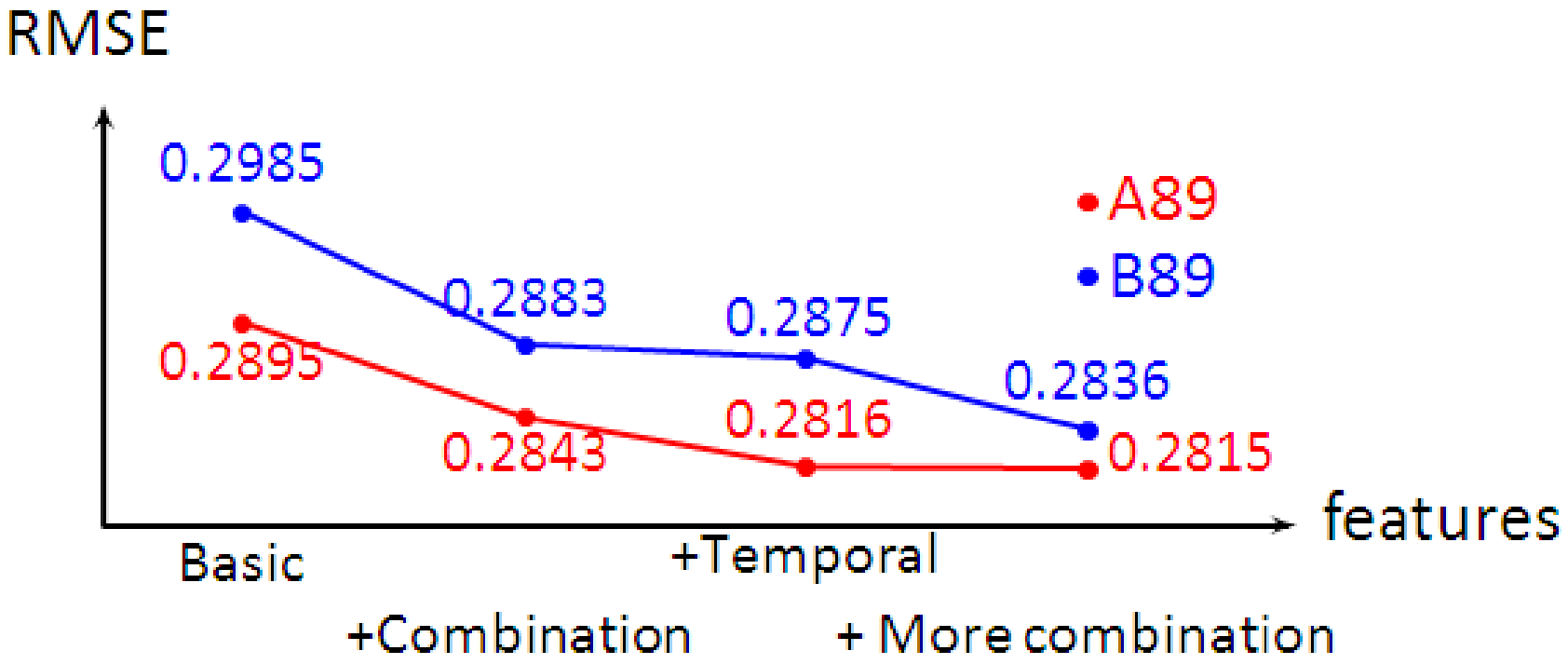
Feature Combinations

- Training size can be large → **nonlinear classifier** (e.g. Kernel SVM) becomes impractical
- **Linear classifier** is preferred, but cannot exploit possible feature dependency
- Feature combination allows a linear classifier to exploit non-linear dependency of features
  - Polynomial mapping (e.g. bigram/trigram features)
  - This turns out to be the key improvement for KDD Cup 2010 but not 2011
- Combining hierarchical information is also useful
  - E.g. (unit name, section, problem name)

Features from Near-by Instances are usually helpful

- One can combine features from similar instances to build a richer model
- What are considered as similar instances?
  - kNN in terms of features
  - Instances close in time
  - Instances close in space
  - …

Results


RMSE
0.2985
0.2883
0.2875
0.2836
0.2895
0.2843
0.2816
0.2815
A89
B89
features
Basic
+Temporal
+Combination
+ More combination

Key Issues for Practical ML/KDD

- Feature Engineering
- Blending and Ensemble The more models the better
- Training and Optimization
- Validation
- Novel ideas
- Smart and motivated personnel with significant amount of time investigated

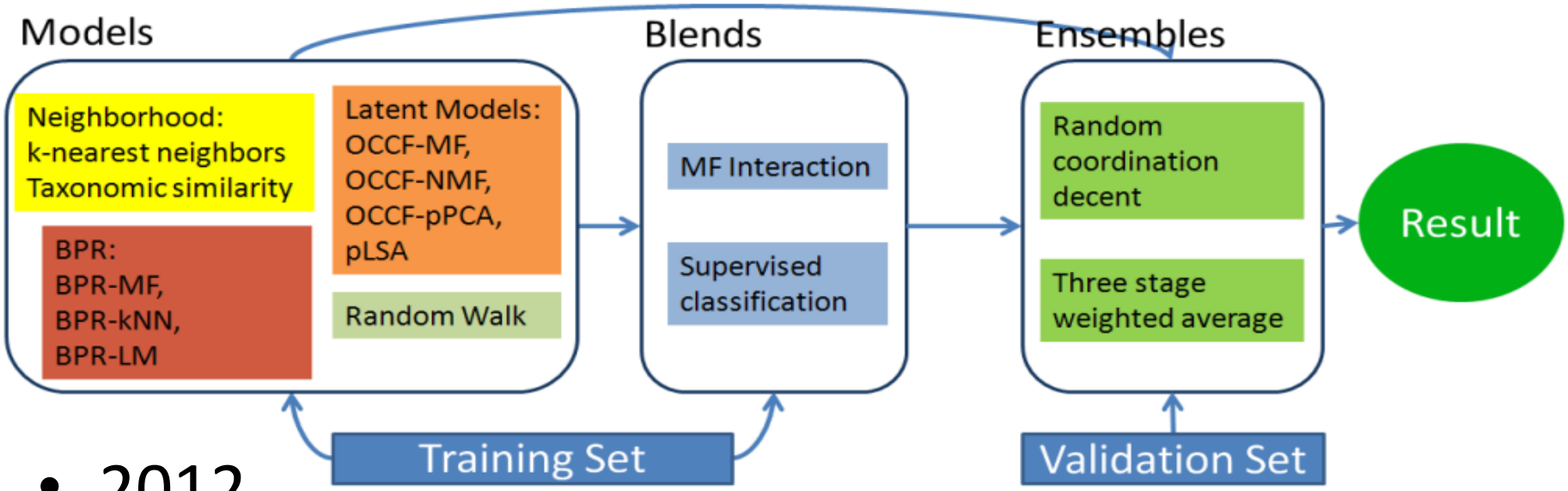
Blending and Ensemble

- Most of the top teams in recent KDD Cups exploit such technique one way of another.
- Blending: combine the results from some models
  - Usually the number of models are not a lot
  - Non-linear methods such as kernel-SVM or neural network can be exploited
- Ensemble: combine the results from blending models and individual models
  - Usually takes a large amount of models
  - Simple linear or voting methods are exploited
  - Be careful, can cause overfitting.

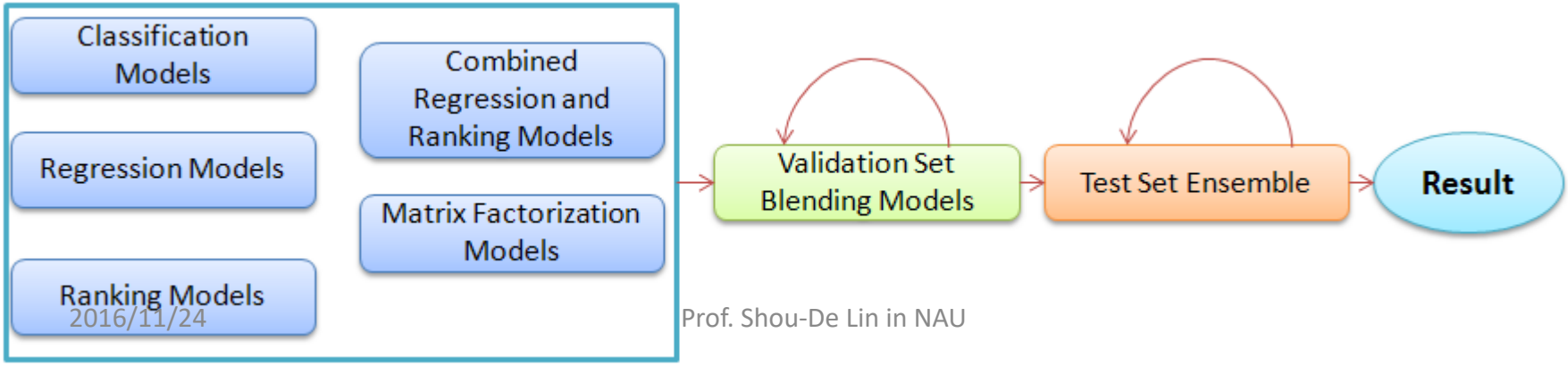
The golden rule: increasing diversity of models is the key to the success blending and ensemble.

Case Study: KDD Cup 2011 and 2012

- 2011



- 2012



Ensemble brings a different mindset to assess the quality of a model

- Does a worse model really has no value?
- A worse model is useful as long as it brings diversity
- A superior model might not be useful if it does not bring diversity.

Glance of Single Model RMSE

model	# used	best	average	worst	contribution
MF	81	22.90	23.92	26.94	**0.3645**
pPCA	2	24.46	24.61	24.75	0.0014
pLSA	7	24.83	25.53	26.09	0.0042
R-Boltz. machine	8	22.80	24.75	26.08	0.0314
k-NN	18	22.79	25.06	42.94	0.0298
regression	10	24.13	28.01	35.14	0.0261

- contribution (**before val.-set blending**): estimated RMSE diff. via leave-the-model-out in test-set blending
- MF: most important (absorbing pPCA)
- residual models: both quite important
- derivative model: individually weak but adds diversity

val.-set blending:
95 models, best 21.36, average 23.53, worst 31.70

Where does the diversity come from?

- Different models
  - Different objective function to optimize
  - Same objective but different optimization method
  - Adding/subtracting some constraints (e.g. MF vs. NMF)
- Different parameters and initialization for the models (not as useful as different models, but can avoid overfitting)
- Different sampling over data or different validation set

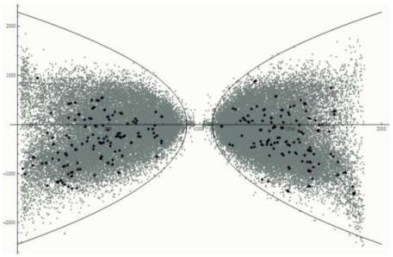
What are the Key Issues in Practical ML/KDD

- Feature Engineering
- Blending and Ensemble
- Validation
- Training and Optimization
- Novel ideas
- Smart and motivated personnel with significant amount of time investigated

Correctly Producing Validation Set is Important

- A validation set is necessary to tune up parameters.
  - Cross Validation (CV) might not be feasible when you have limited time
- How such set is chosen can significantly affect the level of overfitting.

Case Study: KDD Cup 2008 (identify potential cancer patients)



- Training set: a set of positive and negative instances (each contains 118 features)
  - Each positive patient contain a set of negative instances (i.e. an ROI in the X-ray) and at least one positive instances.
  - ALL instances in a negative patient are negative
  - It's a multi-instance classification problem.
- Random division for CV:
  - training: 90%, testing: 72% → significantly overfitting
- Patient-based CV:
  - training: 80%, testing: 77%

Case Study: KDDCUP 2011 track 2- A representative validation set

- In KDD Cup 2011 track 2, we need to do sampling to create validation data (3 positive and 3 negative) from ratings
  - Random sample → not good enough
  - We sample several different sets, and test on a variety algorithms and choose one that obtain similar improvement ratio with the testing results
  - It turned out our validation set is 0.5% lower than that of the testing results, but very consistent.

What are the Key Issues in Practical ML/KDD

- Feature Engineering
- Blending and Ensemble
- Validation
- Training and Optimization
- Novel ideas
- Smart and motivated personnel with significant amount of time investigated

Training and Optimization

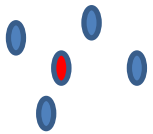
- Avoid overfitting is very important
  - Add regularization terms
  - Occam's Razor: Linear > non-linear, few features in more instances > more features in fewer instances
- Different optimization methods can lead to different results given the same objective function, e.g. in KDDCUP 2011, we have
  - SGD, RCD, Greedy, Simulated annealing methods, and each yields different performance.
  - We finally blending all of them to get better results.

What are the Key Issues in Practical ML/KDD

- Feature Engineering
- Blending and Ensemble
- Validation
- Training and Optimization
- Novel ideas
- Smart and motivated personnel with significant amount of time investigated

Ideas that are useful **sometimes**

- Add randomness into the model (e.g. random restart), and then average the results can avoid overfitting (we found this very useful in 2011).
- Adaptive Learning: during learning, it is generally helpful to decrease the learning rate when the improvement becomes smaller (0.1%~0.6% improvement in KDD Cup 2011).
- Residual-based approaches
  - The basic model is first trained for prediction, and then its residuals are used as the inputs to train a residual-based model. During prediction, a similar procedure is applied and the outputs from both models are summed up as the final output.
- Apply kNN idea in predicting ratings



Relevant Publication

- Chun-Liang Lee “Combination of feature engineering and ranking models for paper-author identification in KDD Cup 2013”, ***Journal of Machine Learning Research , 2015***
- Wei-Sheng Chin et al, “Effective string processing and matching for author disambiguation” ***Journal of Machine Learning Research , 2014***
- Kuan-Wei Wu, et al, “A Two-Stage Ensemble of Diverse Models for Advertisement Ranking in KDD Cup 2012”, **KDD Cup Workshop 2012**
- Todd G. McKenzie, et al. “Novel Models and Ensemble Techniques to Discriminate Favorite Items from Unrated Ones for Personalized Music Recommendation” ***Journal of Machine Learning Research*** **Workshop and Conference Proceedings, 2012**
- Po-Lung Chen, et al “A Linear Ensemble of Individual and Blended Models for Music Rating Prediction” ***Journal of Machine Learning Research*** **Workshop and Conference Proceedings, 2012**
- Hsiang-Fu Yu, et al “Feature engineering and classifier ensemble for KDD Cup 2010”, to appear in **Journal of Machine Learning Research, Workshop and Conference Proceedings**, 2011
- Hung-Yi Lo, et al. *“An Ensemble of Three Classifiers for KDDCup 2009: Expanded, Linear Model, Heterogeneous Boosting, and Selective Naïve Bayes* ***“ Journal of Machine Learning Research*** **Workshop and Conference Proceedings** *2009*
- Hung-Yi Lo et al. "Learning to Improve Area-Under-FROC for Imbalanced Medical Data Classification Using an Ensemble Method" **ACM SIGKDD Explorations** Vol10, Issue 2, 2008

推薦系統方法

林守德教授

CSIE, NTU

sdlin@csie.ntu.edu.tw

Agenda

- Recommender System: an overview
- Introducing various recommendation models
- Evaluating a recommender system
- Advanced issues in recommendation
- Tools or Library for building a recommender system

What is the Recommendation System(RS)?

- A recommendation system suggests potentially favored items or contents to users
  - Users may be recommended new items previously unknown to them.
  - The system can find the opposite (e.g.: items the users do not like)

推薦系統就是那個「心魔」

Values of Recommendation Systems

- For service providers
  - Improve trust and customer loyalty
  - Increase sales, click trough rates, etc.
  - Opportunities for promotion
- For customers
  - New interesting items being identified
  - Narrow down the possible choices

NETFLIX

2/3 of the movies watched are recommended

recommendations generate 38% more click-throughs

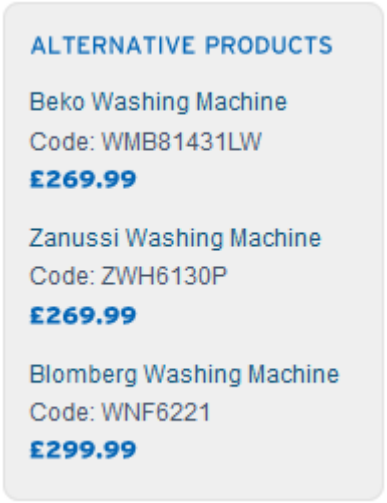
amazon

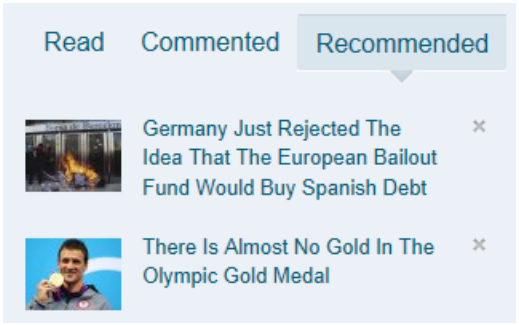
35% sales from recommendations

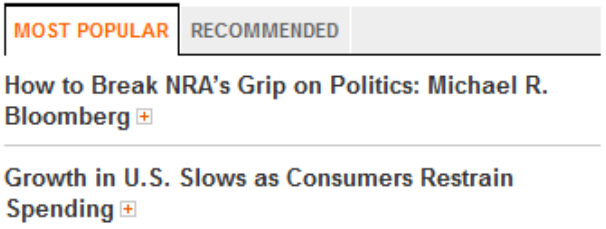
cs.
choicestream.

28% of the people would buy more music if they found what they liked

Recommending products, services, and news

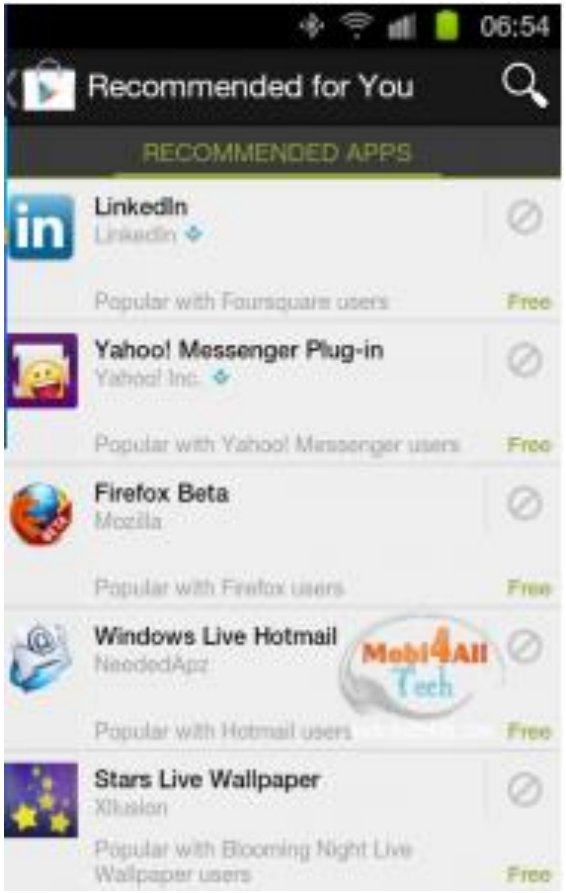


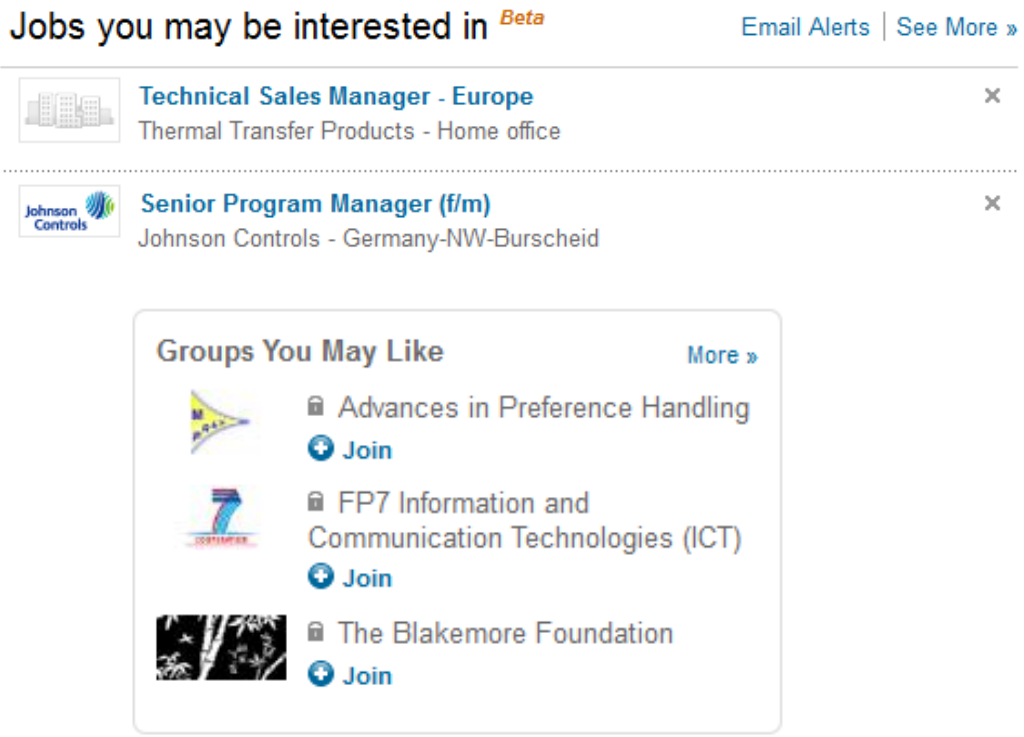




From Dietmar Jannach, Markus Zanker and Gerhard Friedrich

Recommending friends, jobs, and photos





From Dietmar Jannach, Markus Zanker and Gerhard Friedrich

The Netflix Challenge （2006～2009）

- 1M prize to improve the prediction accuracy by 10%

KDD Cup 2011: Yahoo Music Recommendation
KDD Cup 2012: Tencent Advertisement Recommendation

What is a good recommender system?

- **Requirement 1: finding items that interests the specific user → personalization**
- Requirement 2: recommending diverse items that satisfy all the possible needs of a specific user → diversity
- Requirement 3: making non-trivial recommendation (Harry Potter 1 ,2 ,3 ,4 → 5) → novelty

Inputs/outputs of a Recommendation System

- Given:
  - Ratings of users to items: rating can be either explicit or implicit, for example
    - Explicit ratings: the level of likeliness of users to items
    - Implicit ratings: the browsing information of users to items
  - User features (e.g. preferences, demographics)
  - Item features (e.g. item category, keywords)
  - User/user relationships
- Predict:
  - Ratings of any user to any item, or
  - Ranking of items to each user
- What to recommend?
  - Highly rated or ranked items given each user

Types of Recommender Systems

- Content-based Recommendation (CBR)
- Collaborative Filtering
- Other solutions
- Advanced Issues

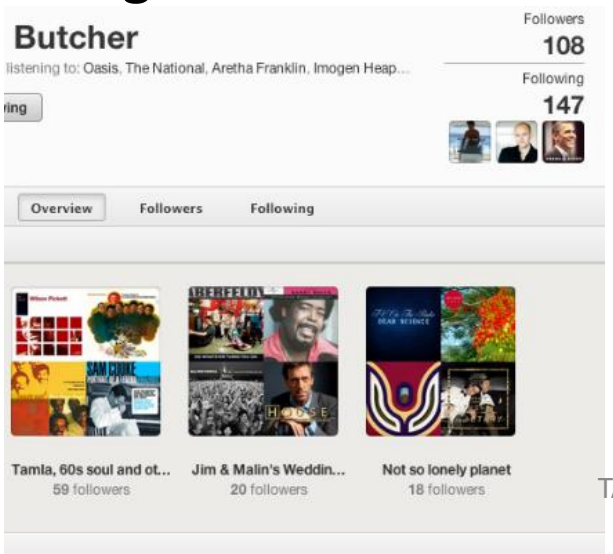
Key Concept of CBR

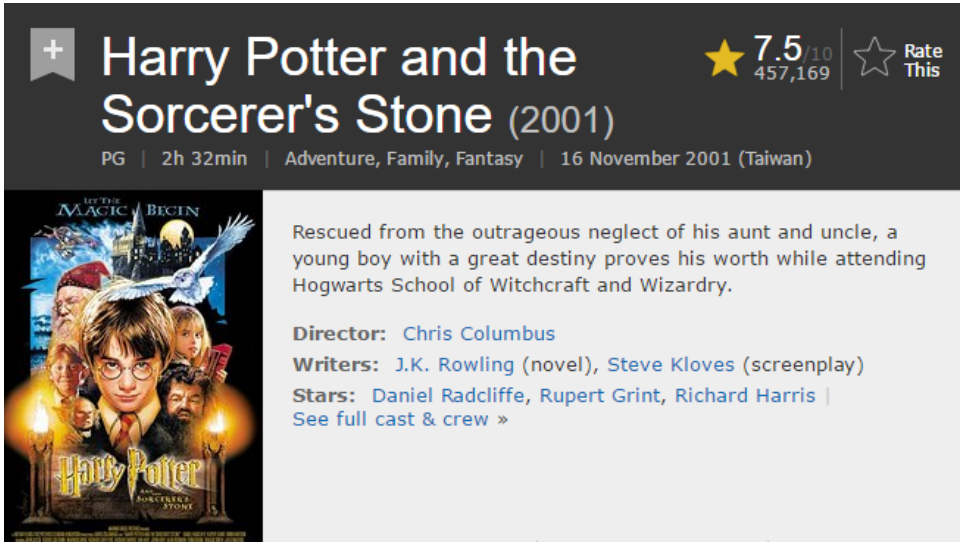
1. Recommend items that are similar to the ones liked by this user in the past

2. Recommend items whose attributes are similar to the users' profile

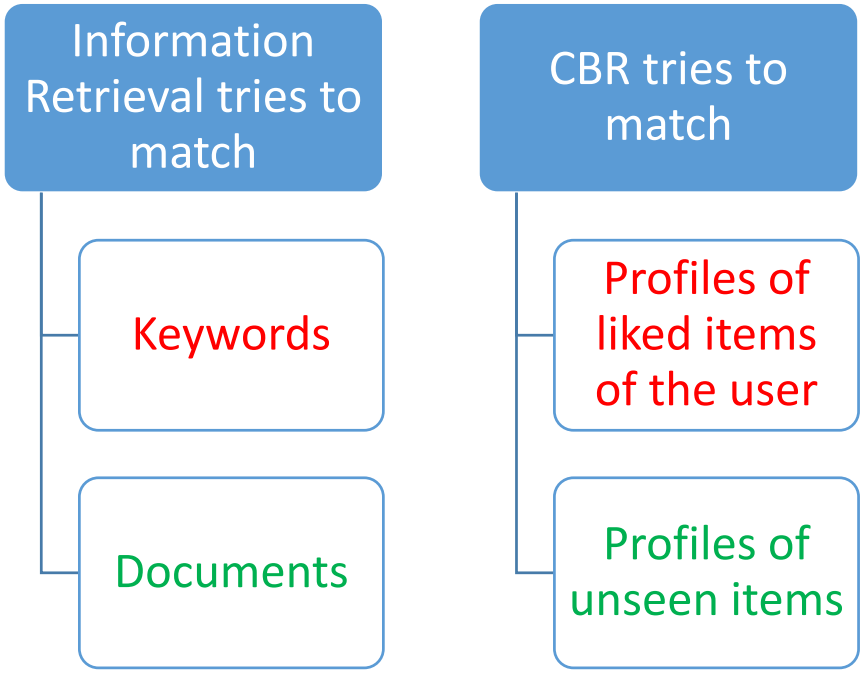
CBR: Requirement

- Required info:
  - information about items (e.g. genre and actors of a movie, or textual description of a book)
  - Information about users (e.g. user profile info, demographic features, etc)
- Optional Info
  - Ratings from users to items





Content-based Recommendation vs. Information Retrieval (i.e. Search Engine)



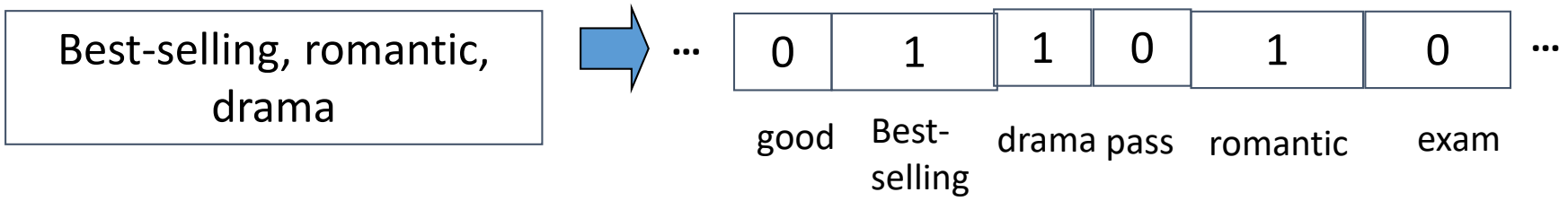
- If we know how keywords can be used to match documents in a search engine, then we will use the same model to match user and item profiles

Vector Space Model

- Represent the keywords of items using a **term vector**
  - Term: basic concept, e.g., keywords to describe an item
  - Each term represents one dimension in a vector
  - N total terms define an n-element terms
  - Values of each term in a vector corresponds to the importance of that term
    - E.g., $d=(x_1,\ldots,x_N)$, x_i is "importance" of term i
- Measure similarity by the vector distances

An Example

- Items or users are represented using a set of keywords.
  - each dimension represents a binary random variable associated to the existence of an indexed term (1:exist, 0: not exist in the document)



Term Frequency and Inverse Document Frequency (TFIDF)

- Since not all terms in the vector space are equally important, we can weight each term using its occurrence probability in the item description
  - ***Term frequency:*** *TF(d,t)*
    - number of times *t* occurs in the item description d
  - ***Inverse document frequency:*** *IDF(t)*
    - to scale down the terms that occur in many descriptions

Normalizing Term Frequency

- n_{ij} represents the number of times a term t_i occurs in a description d_j . Tf_{ij} can be normalized using the total number of terms in the document.

$$tf_{ij} = \frac{n_{ij}}{\sum_k n_{kj}}$$

- Another normalization:

$$tf_{ij} = \frac{n_{ij}}{\max n_{kj}}$$

Inverse Document Frequency

- IDF seeks to scale down the coordinates of terms that occur in many item descriptions.
  - For example, some stop words (the, a, of, to, and…) may occur many times in a description. However, they should not be considered as important representation feature.
- IDF of a term t_i : $idf_i = \log(\frac{N}{df_i} + 1)$

,where df_i (document frequency of term t_i) is the number of descriptions in which t_i occurs.

Term Frequency and Inverse Document Frequency

- TF-IDF weighting : weight(t,d)=TF(t,d)*IDF(t)
  - Common in doc →high tf → high weight
  - Rare in collection → high idf→ high weight
- weight w_{ij} of a term t_i in a document d_j

$$w_{ij} = tf_{ij} * (\log \frac{N}{df_j} + 1)$$

Computing TF-IDF -- An Example

Given an item description contains terms with given frequencies:

$$tf_{ij} = \frac{n_{ij}}{\max n_{kj}}$$

A: best-selling (3), B: Romantic (2), C: Drama (1)

Assume collection contains 10,000 items and

document frequencies of these terms are:

best-selling(50), Romantic (1300), Drama (250)

Then:

A: tf = 3/3; idf = $\log_2$(10000/50+1) = 7.6; tf-idf = 7.6

B: tf = 2/3; idf = $\log_2$ (10000/1300+1) = 2.9; tf-idf = 2.0

C: tf = 1/3; idf = $\log_2$ (10000/250+1) = 5.3; tf-idf = 1.8

The vector-space presentation of this item using A, B, and C is (7.6,2.0,1.8)

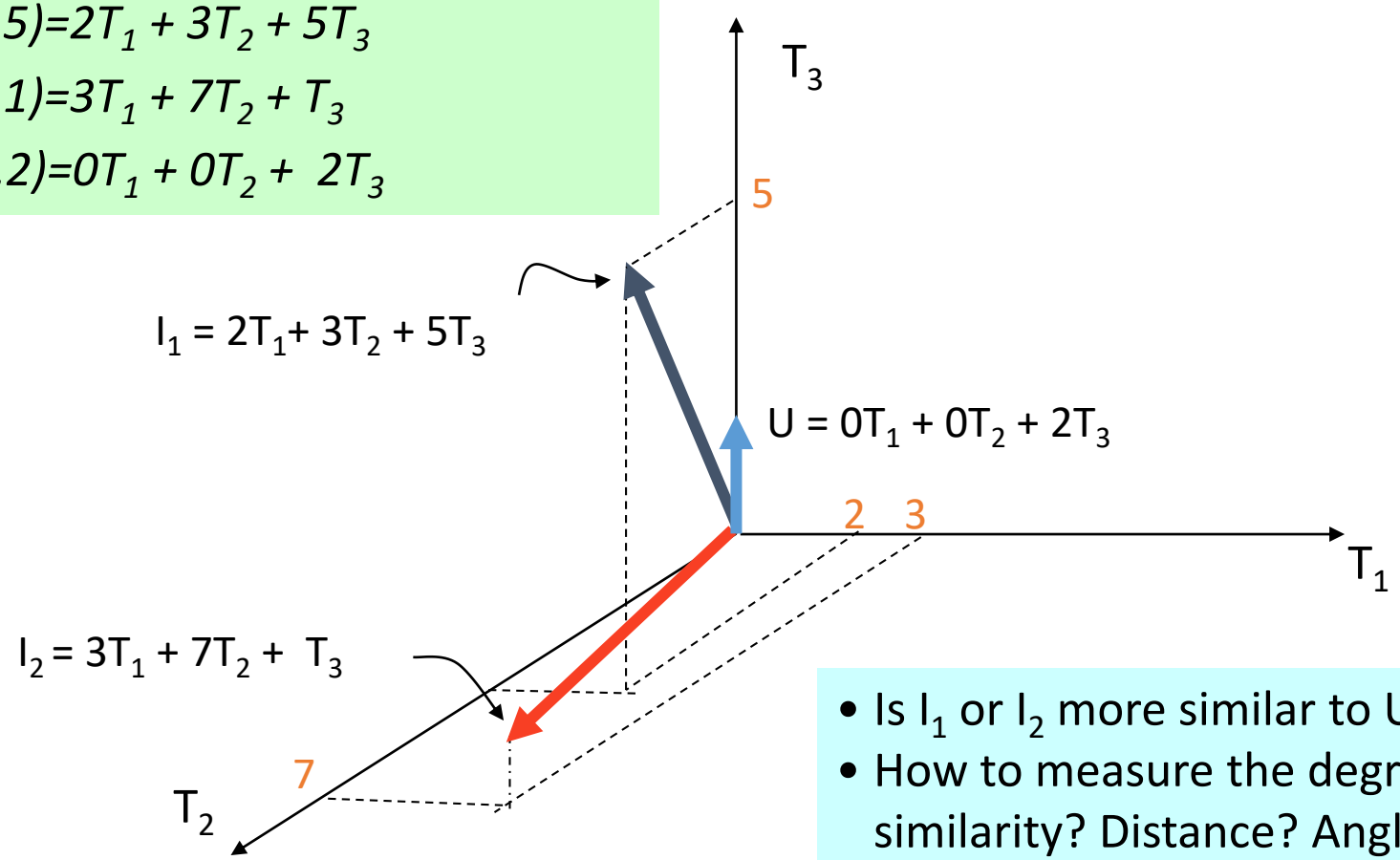
Graphic Representation

Example:

$I_1 = (2,3,5) = 2T_1 + 3T_2 + 5T_3$

$I_2 = (3,7,1) = 3T_1 + 7T_2 + T_3$

$U = (0,0,2) = 0T_1 + 0T_2 + 2T_3$



- Is I_1 or I_2 more similar to U?
- How to measure the degree of similarity? Distance? Angle? Projection?

Measure Similarity Between two item descriptions

- $D=\{d_1,d_2,\ldots d_n\}$
- $Q=\{q_1,q_2,\ldots q_n\}$ (the q_x value is 0 if a term is absent)
- Dot product similarity: $Sim(D,Q)=d_1*q_1+d_2*q_{2+}\ldots+d_n*q_n$
- Cosine Similarity (or normalized dot product):

sim(D,Q)=

$$sim(Q,D)=\frac{d_1*q_1+\ldots+d_n*q_n}{\sqrt{\sum_{j=1}^{N}d_j^{\,2}*\sum_{j=1}^{N}q_j^{\,2}}}$$

Content-based Recommendation: an example from Alexandros Karatzoglou

- A customer buys the book: "*Building data mining applications for CBM*"
- 7 Books are possible candidates for a recommendation:
  1. *Accelerating Customer Relationships: Using CRM and Relationship Technologies*
  2. *Mastering Data Mining: The Art and Science of Customer Relationship Management*
  3. *Data Mining Your Website*
  4. *Introduction to marketing*
  5. *Consumer behaviour*
  6. *Marketing research, a handbook*
  7. *Customer knowledge management*

Using binary existence of words

COUNT	a	Accelerating	and	applications	art	behavior	Building	Consumer	CRM	customer	data	for	Handbook	Introduction	Knowledge	Management	Marketing	Mastering	mining	of	relationship	Research	science	technology	the	to	using	website	your
Building data mining applications for CRM				1			1		1		1	1							1										
Accelerating customer relationships: using CRM and relationship technologies		1	1						1	1											2			1			1		
Mastering Data Mining: the art and science of Customer Relationship Management			1		1					1	1					1		1	1	1	1		1		1				
Data Mining your website											1								1									1	1
Introduction to Marketing														1			1									1			
Consumer behavior						1		1																					
Marketing Research: a Handbook	1												1				1					1							
Customer Knowledge Management										1					1	1											25		

Using TFIDF

TFIDF Normed Vectors	a	Accelerating	and	applications	art	behavior	Building	Consumer	CRM	customer	data	for	Handbook	Introduction	Knowledge	Management	Marketing	Mastering	mining	of	relationship	Research	science	technology	the	to	using	website	your
Building data mining applications for CRM				0.502			0.502		0.344		0.251	0.502							0.251										
Accelerating customer relationships: using CRM and relationship technologies		0.432	0.296						0.296	0.216											0.468			0.432			0.432		
Mastering Data Mining: the art and science of Customer Relationship Management			0.256		0.374					0.187	0.187					0.256		0.374	0.187	0.374	0.256		0.374		0.374				
Data Mining your website											0.316								0.316									0.632	0.632
Introduction to Marketing														0.636			0.436									0.636			
Consumer behavior						0.707		0.707																					
Marketing Research: a Handbook	0.537												0.537				0.368					0.537							
Customer Knowledge Management										0.381					0.736	0.522													

Pros and Cons for CBR

Pros:

- user independence – does not need data from other users
- transparency – easily explainable
- new items can be easily incorporated (assuming content available)

Cons:

- Difficult to apply to domain where feature extraction is an inherent problem (e.g. multimedia data)
- Keywords might not be sufficient: Items represented by the same set of features are indistinguishable
- Cannot deal with new users
- Redundancy: should we show items that are too similar?

Types of Recommender Systems

- Content-based Recommendation (CBR)
- Collaborative Filtering
- Other solutions
- Advanced Issues

Collaborative Filtering (CF)

- CF is the most successful and common approach to generate recommendations
  - used in Amazon, Netflix, and most of the e-commerce sites
  - many algorithms exist
  - General and can be applied to many domains (book, movies, DVDs, ..)
- Key Idea
  - Recommended the favored items of people who are 'similar' to you
  - Need to collect the taste (implicit or explicit) of other people ➔ that's why we call it 'collaborative'

Mathematic Form of CF

- Given: some ratings from users to items
- Predict: unknown ratings of users to items

Explicit ratings	Item1	Item2	Item3	Item4
User1	?	3	?	?
User2	1	?	?	5
User3	?	4	?	2
User4	3	?	3	?

Implicit ratings	Item1	Item2	Item3	Item4
User1	?	1	?	?
User2	1	?	?	1
User3	?	1	?	1
User4	1	?	1	?

CF models

- Memory-based CF
  - User-based CF
  - Item-based CF
- Model-based CF

User-Based Collaborative Filtering

- Finding users $N(u)$ most similar to user u (neighborhood of u)
  - Assign $N(u)$'s rating as u's rating
- Prediction
  - $r_{u,i} = \overline{r}_u + \frac{\sum_{v \in N(u)} sim(u,v)(r_{v,i} - \overline{r}_v)}{\sum_{v \in N(u)} sim(u,v)}$
  - $\overline{r}_u$: Average of ratings of user u
- Problem: Usually users do not have many ratings; therefore, the similarities between users may be unreliable

Example of User-Based CF

$$\bar{r}_{John} = \frac{3+0+3+3}{4} = 2.25$$

$$\bar{\boldsymbol{r}}_{\boldsymbol{Joe}} = \frac{5+4+0+2}{4} = \mathbf{2.75}$$

$$\bar{r}_{Jill} = \frac{1+2+4+2}{4} = 2.25$$

$$\bar{r}_{Jane} = \frac{3+1+0}{3} = 1.33$$

$$\bar{\boldsymbol{r}}_{\boldsymbol{Jorge}} = \frac{2+2+0+1}{4} = \mathbf{1.25}$$

$$sim(Jane, John) = \frac{3\times3+1\times3+0\times3}{\sqrt{3^2+1^2+0^2}\sqrt{3^2+3^2+3^2}} = 0.73$$

$$sim(Jane, \boldsymbol{Joe}) = \frac{3\times5+1\times0+0\times2}{\sqrt{3^2+1^2+0^2}\sqrt{5^2+0^2+2^2}} = \mathbf{0.88}$$

$$sim(Jane, Jill) = \frac{3\times1+1\times4+0\times2}{\sqrt{3^2+1^2+0^2}\sqrt{1^2+4^2+2^2}} = 0.48$$

$$sim(Jane, \boldsymbol{Jorge}) = \frac{3\times2+1\times0+0\times1}{\sqrt{3^2+1^2+0^2}\sqrt{2^2+0^2+1^2}} = \mathbf{0.84}$$

$$r_{Jane,Aladdin} = 1.33 + \frac{0.88(4-2.75)+0.84(2-1.25)}{0.88+0.84} = 2.33$$

- To predict $r_{Jane,Aladdin}$ using cosine similarity
  - Neighborhood size is 2

Rating	Lion King	Aladdin	Mulan	Anastasia
John	3	0	3	3
Joe	5	4	0	2
Jill	1	2	4	2
Jane	3	?	1	0
Jorge	2	2	0	1

Item-Based Collaborative Filtering

- Similarity is defined between two items
- Finding items $N(i)$ most similar to item i (neighborhood of i)
  - Assign $N(i)$'s rating as i's rating
- Prediction
  - $r_{u,i} = \overline{r}_i + \frac{\sum_{j \in N(i)} sim(i,j)(r_{u,j} - \overline{r}_j)}{\sum_{j \in N(i)} sim(i,j)}$
  - $\overline{r}_i$: Average of ratings of item i
- Items usually have more ratings from many users and the similarities between items are more stable

Example of Item-Based CF

$$\bar{r}_{\textbf{Lion King}} = \frac{3+5+1+3+2}{5} = \mathbf{2.8}$$

$$\bar{r}_{Aladdin} = \frac{0+4+2+2}{4} = 2$$

$$\bar{r}_{Mulan} = \frac{3+0+4+1+0}{5} = 1.6$$

$$\bar{r}_{\textbf{Anastasia}} = \frac{3+2+2+0+1}{5} = \mathbf{1.6}$$

$$sim(Aladdin, \textbf{Lion King}) = \frac{0\times3+4\times5+2\times1+2\times2}{\sqrt{0^2+4^2+2^2+2^2}\sqrt{3^2+5^2+1^2+2^2}} = \mathbf{0.84}$$

$$sim(Aladdin, Mulan) = \frac{0\times3+4\times0+2\times4+2\times0}{\sqrt{0^2+4^2+2^2+2^2}\sqrt{3^2+0^2+4^2+0^2}} = 0.32$$

$$sim(Aladdin, \textbf{Anastasia}) = \frac{0\times3+4\times2+2\times2+2\times1}{\sqrt{0^2+4^2+2^2+2^2}\sqrt{3^2+2^2+2^2+1^2}} = \mathbf{0.67}$$

$$r_{Jane,Aladdin} = 2 + \frac{0.84(3-2.8)+0.67(0-1.6)}{0.84+0.67} = 2.33$$

- To predict $r_{Jane,Aladdin}$ using cosine similarity
  - Neighborhood size is 2

Rating	Lion King	Aladdin	Mulan	Anastasia
John	3	0	3	3
Joe	5	4	0	2
Jill	1	2	4	2
Jane	3	?	1	0
Jorge	2	2	0	1

Quiz: 推薦系統之阿拉丁神燈

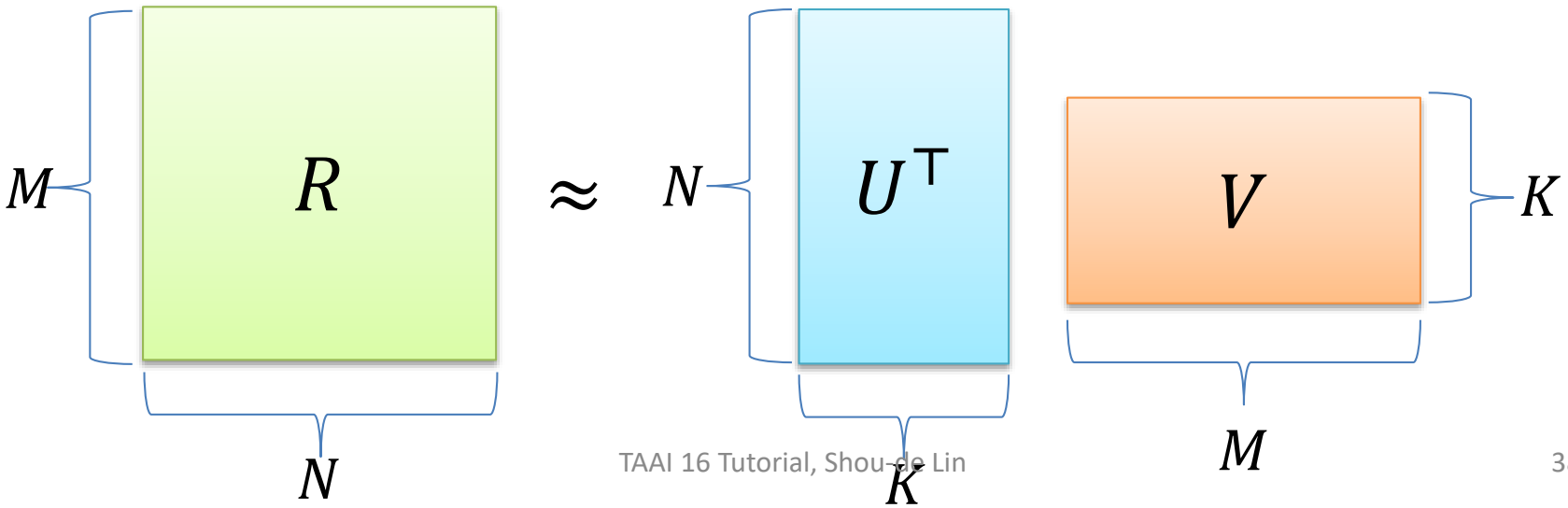
- 小胖撿到了一個神燈，他摩擦了神燈，神燈巨人答應他一個願望，小胖說：我想要蔡10陪我共進晚餐。神燈巨人說沒問題，繃的一聲，就把蔡10，蔡13都變了出來。小胖說：我沒有說要找蔡13啊？神燈巨人說：我的推薦系統認為你也可能對蔡依珊感興趣。
- 請問，神燈巨人內部是跑哪種推薦系統

CF models

- Memory-based CF
  - User-based CF
  - Item-based CF
- Model-based CF
  - Matrix Factorization

Matrix Factorization (MF)

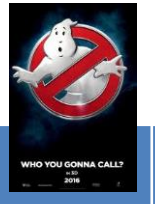
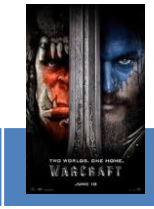
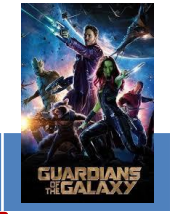
- Given a matrix $R \in \mathbb{R}^{N \times M}$, we would like to find two matrices $U \in \mathbb{R}^{K \times N}, V \in \mathbb{R}^{K \times M}$ such that $U^\top V \approx R$
  - $K \ll \min\{N, M\}$ ➔ we assume R of small rank K
  - A low-rank approximation method
  - Earlier works (before 2007 ~ 2009) call it singular value decomposition (SVD)



Matrix factorization

$$R = U^{\mathsf{T}} V$$

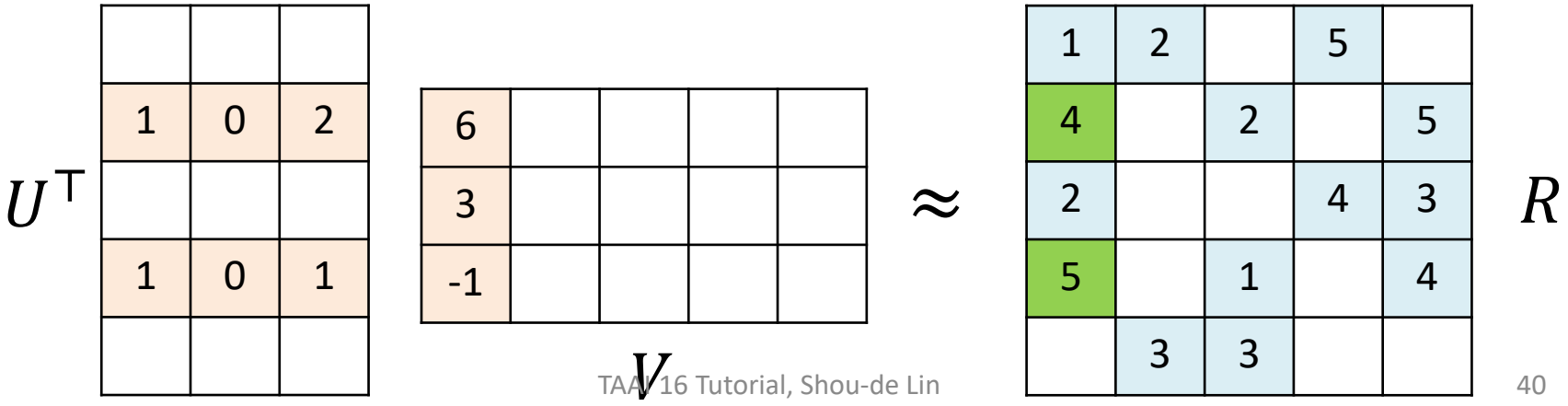
U_k	Dim1	Dim2
Alice	0.4	0.3
Bob	-0.4	0.3
Mary	0.7	-0.6
Sue	0.3	0.9

V_k^{T}					
Dim1	-0.4	-0.7	0.6	0.4	0.5
Dim2	0.8	-0.6	0.2	0.2	-0.3

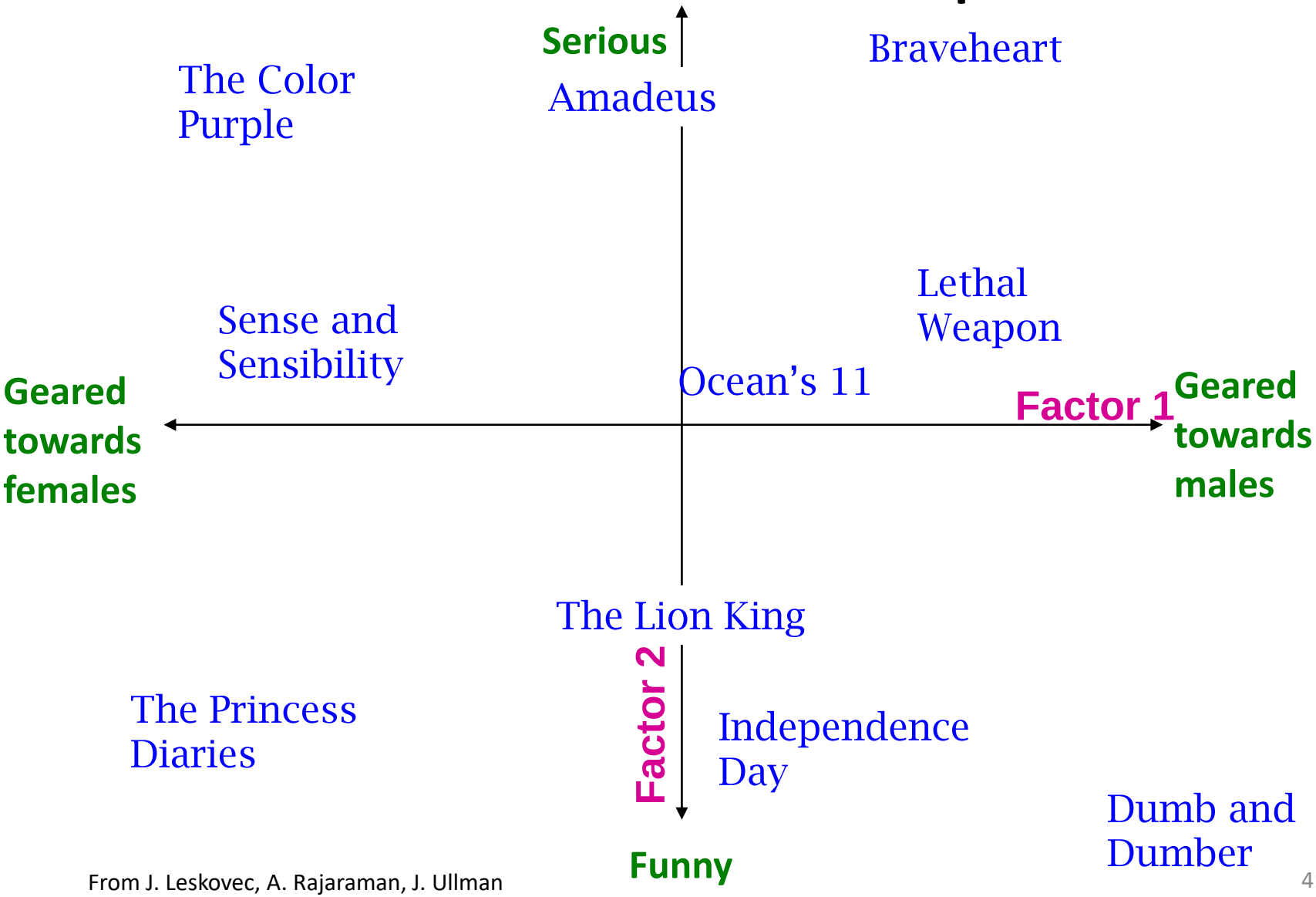
Rating (Alice to Harry Potter)=0.4*0.4+0.3*0.2

MF as an Effective CF Realization

- We find two matrices U, V to approximate R
  - Missing entry R_{ij} can be predicted by $U_i^\top V_j$
- Entries in column U_i (or V_j)represent the latent factor (i.e. rating patterns learned from data) of user i (or item j)
- If two users have similar latent factors, then they will give similar ratings to all items
- If two items have similar latent factor, then the corresponding rating for all users are similar



Latent Factor Examples



From J. Leskovec, A. Rajaraman, J. Ullman

How to Train an MF Model (1/2)

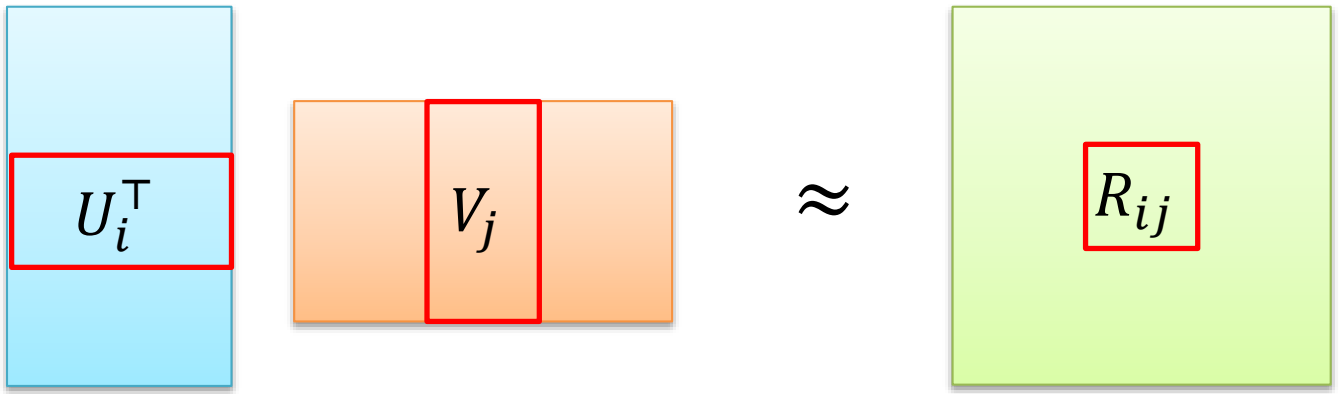
- MF as a Minimization problem

$$\arg\min_{U,V} \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{M}\delta_{ij}\left(U_i^{\top}V_j - R_{ij}\right)^2 + \frac{\lambda_U}{2}\|U\|_F^2 + \frac{\lambda_V}{2}\|V\|_F^2$$

Squared error

Regularization

  - $\|U\|_F^2 = \sum_{i=1}^{N}\|U_i\|_2^2 = \sum_{i=1}^{N}\sum_{k=1}^{K} U_{ki}^2$: squared Frobenius norm
  - $\delta_{ij} \in \{0,1\}$: rating R_{ij} is observed in R



How to Train an MF Model (2/2)

- MF with bias terms

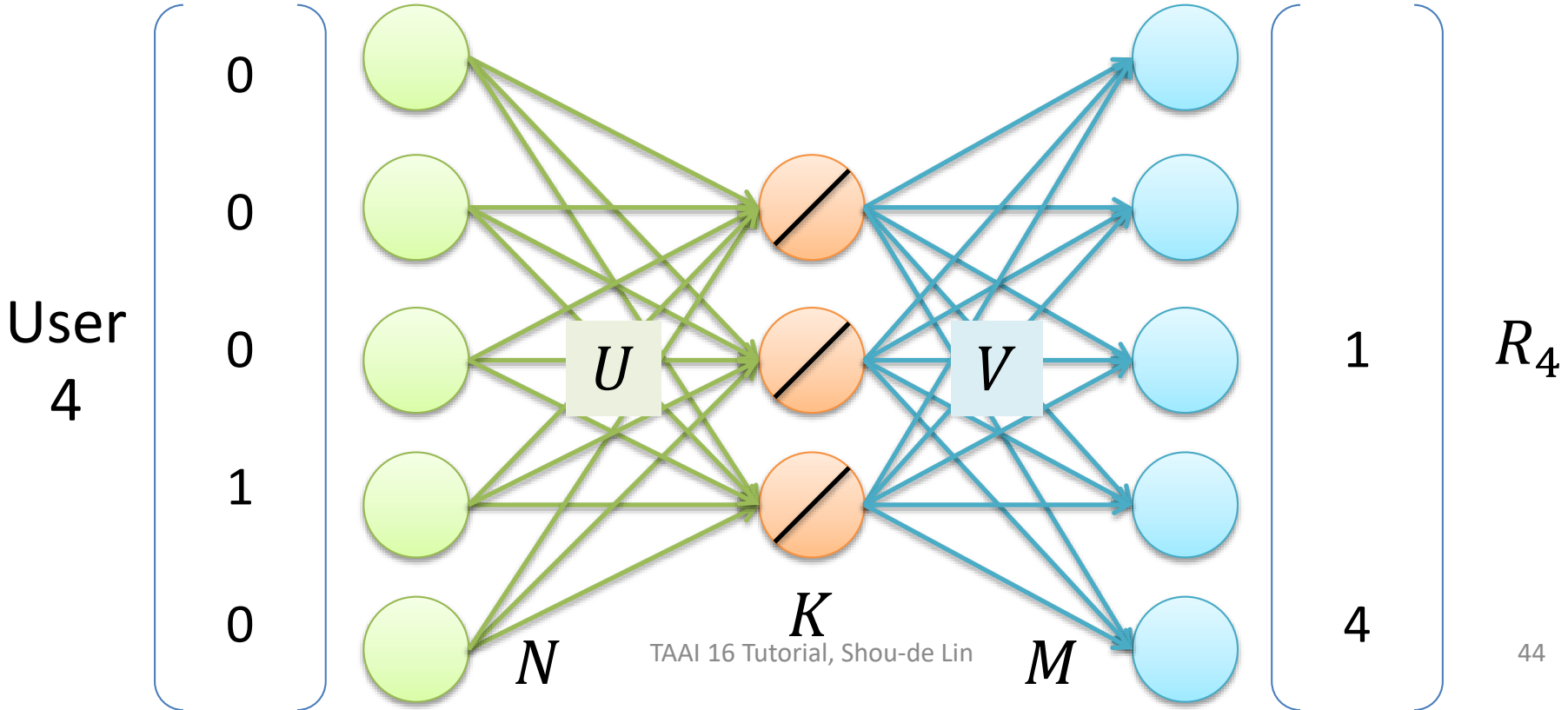
$$\arg\min_{U,V} \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{M}\delta_{ij}\left(U_i^{\top}V_j + b_i + c_j + \mu - R_{ij}\right)^2 + \frac{\lambda_U}{2}\|U\|_F^2 + \frac{\lambda_V}{2}\|V\|_F^2 + \frac{\lambda_b}{2}\|b\|_2^2 + \frac{\lambda_c}{2}\|c\|_2^2 + \frac{\lambda_\mu}{2}\mu^2$$

  - b: rating mean vector for each user
  - c: rating mean vector for each item
  - μ: overall mean of all ratings
- Some MF extension works omit the bias terms

Koren, Yehuda, Robert Bell, and Chris Volinsky.
"Matrix factorization techniques for recommender systems."
Computer 42.8 (2009): 30-37.

MF as Neural Network (NN)

- Shallow NN with identity activation function
  - N input neurons: user i as one-hot encoding
  - M output neurons: row i in rating matrix R



MF as Probabilistic Graphical Model (PGM)

- Bayesian network with normal distributions
- Maximum a posteriori (MAP)

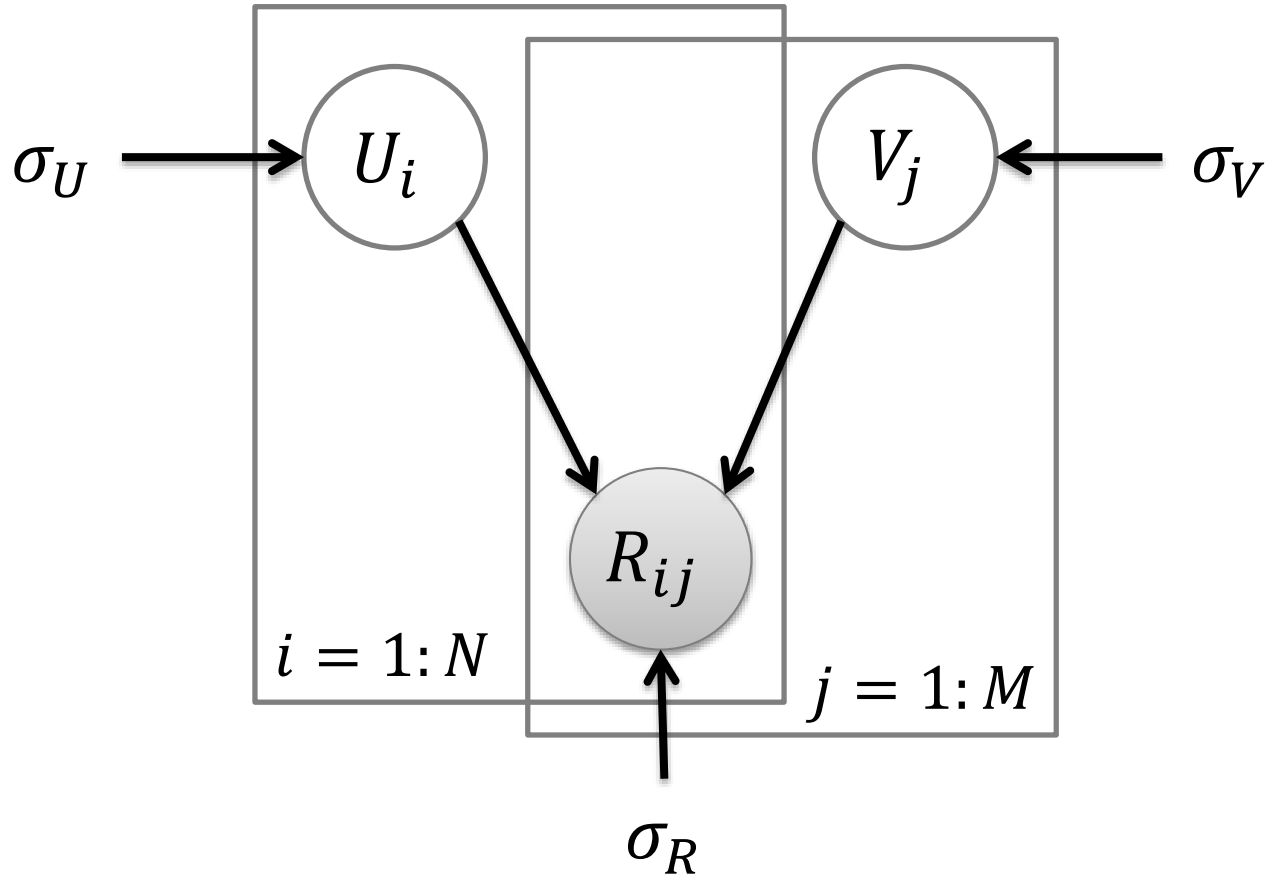
$$\arg\max_{U,V} \prod_{i=1}^{N} \prod_{j=1}^{M} \mathcal{N}\left(R_{ij} \middle| U_i^{\top} V_j, \sigma_R^2\right)^{\delta_{ij}} \prod_{i=1}^{N} \mathcal{N}\left(U_i | 0, \sigma_U^2 I\right) \prod_{j=1}^{M} \mathcal{N}\left(V_j | 0, \sigma_V^2 I\right)$$

Likelihood
(normal distribution)

Zero-mean spherical Gaussian prior
(multivariate normal distribution)

Probabilistic Matrix Factorization (PMF)

- PGM viewpoint of MF



A. Mnih and R. R. Salakhutdinov, "Probabilistic matrix factorization," in Proc. of NIPS, 2008, pp. 1257–1264.

PMF vs. MF

- PMF is equivalent to MF
  - $\lambda_U = \frac{\sigma_R^2}{\sigma_U^2}, \lambda_V = \frac{\sigma_R^2}{\sigma_V^2}$
  - C: terms irrelevant to U, V
  - $l2$-norm regularization = zero-mean spherical Gaussian prior
  - PMF can be easily incorporated into other PGM models

$$\arg\max_{U,V} \prod_{i=1}^{N}\prod_{j=1}^{M} \mathcal{N}\left(R_{ij}|U_i^\top V_j, \sigma_R^2\right)^{\delta_{ij}} \prod_{i=1}^{N} \mathcal{N}(U_i|0, \sigma_U^2 I) \prod_{j=1}^{M} \mathcal{N}(V_j|0, \sigma_V^2 I)$$

Take $-\sigma_R^2 \log x$ ↓

Take $e^{-\frac{1}{\sigma_R^2}x}$ ↑

$$\arg\min_{U,V} \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} \delta_{ij}\left(U_i^\top V_j - R_{ij}\right)^2 + \frac{\lambda_U}{2}\|U\|_F^2 + \frac{\lambda_V}{2}\|V\|_F^2 + C$$

Learning in MF

- Stochastic gradient descent (SGD)
- Alternating least squares (ALS)
- Variational expectation maximization (VEM)

Stochastic Gradient Descent (SGD) (1/2)

- Gradient descent: updating variables based on the direction of negative gradients
  - SGD updates variables instance-wise
- Let L be the objective function

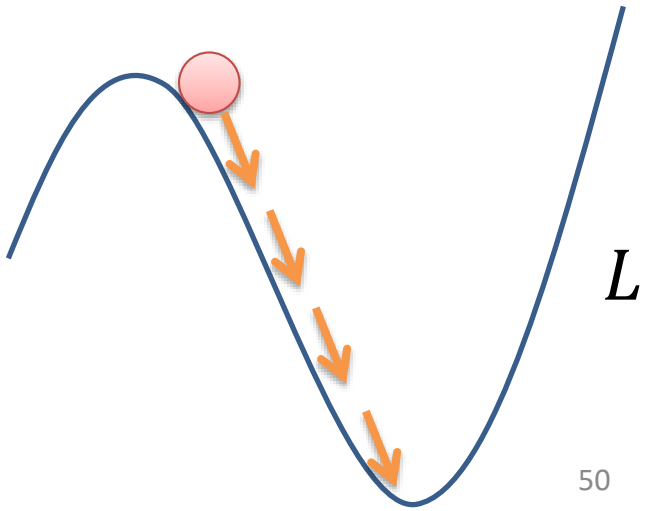
$$L = \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{M}\delta_{ij}\left(U_i^{\top}V_j - R_{ij}\right)^2 + \frac{\lambda_U}{2}\|U\|_F^2 + \frac{\lambda_V}{2}\|V\|_F^2$$

  - Objective function for each training rating

$$L_{ij} = \frac{1}{2}\left(U_i^{\top}V_j - R_{ij}\right)^2 + \frac{\lambda_U}{2}\|U_i\|_2^2 + \frac{\lambda_V}{2}\|V_j\|_2^2$$

Stochastic Gradient Descent (SGD) (2/2)

- Gradient
  - $-\frac{\partial L_{ij}}{\partial U_i} = \left(U_i^\top V_j - R_{ij}\right)V_j + \lambda_U U_i$
  - $-\frac{\partial L_{ij}}{\partial V_j} = \left(U_i^\top V_j - R_{ij}\right)U_i + \lambda_V V_j$
- Update rule
  - $U_i \leftarrow U_i - \eta \frac{\partial L_{ij}}{\partial U_i}$
  - $V_j \leftarrow V_j - \eta \frac{\partial L_{ij}}{\partial V_j}$
  - η: learning rate or step size



Matrix factorization: Stopping criteria

- When do we stop updating?
  - Improvement drops (e.g. <0)
  - Reached small error
  - Achieved predefined # of iterations
  - No time to train anymore

Alternating Least Squares (ALS)

- Stationary point: zero gradient
  - We can find the closed-form solution of U with V fixed, and vise versa
- Zero gradient
  - $\frac{\partial L}{\partial U_i} = \sum_{j=1}^{M} \delta_{ij} V_j \left(U_i^\top V_j - R_{ij} \right) + \lambda_U U_i = 0$
  - $\frac{\partial L}{\partial V_j} = \sum_{i=1}^{N} \delta_{ij} U_i \left(U_i^\top V_j - R_{ij} \right) + \lambda_V V_j = 0$
- Closed-form solution i.e. update rule
  - $U_i = \left(\sum_{j=1}^{M} \delta_{ij} V_j V_j^\top + \lambda_U I \right)^{-1} \left(\sum_{j=1}^{M} \delta_{ij} R_{ij} V_j \right)$
  - $V_j = \left(\sum_{i=1}^{N} \delta_{ij} U_i U_i^\top + \lambda_V I \right)^{-1} \left(\sum_{i=1}^{N} \delta_{ij} R_{ij} U_i \right)$

SGD vs. ALS

$$\arg\min_{U,V} \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{M}\delta_{ij}\left(U_i^{\top}V_j - R_{ij}\right)^2 + \frac{\lambda_U}{2}\|U\|_F^2 + \frac{\lambda_V}{2}\|V\|_F^2$$

- SGD
  - It is easier to develop MF extensions since we do not require the closed-form solutions
- ALS
  - We are free from determine learning rate η
  - It allows parallel computing
- Drawback for both
  - Regularization parameters λ need careful tuning using validation → we have to run MF for multiple times
  - Variational-EM (VEM) learns regularization parameters

Extensions of MF

- Matrix R can be factorized into $U^\top U, U^\top VU, UVW, \ldots$
- SVD++
  - MF with implicit interactions among items
- Non-negative MF (NMF)
  - non-negative entries for the factorized matrices
- Tensor factorization (TF), Factorization machines (FM)
  - Additional features involved in learning latent factors
- Bayesian PMF (BPMF)
  - Further modeling distributions of PMF parameters θ
- Poisson factorization (PF)
  - PMF normal likelihood replaced with Poisson likelihood to form a probabilistic nonnegative MF

Applications of MF

- Recommender systems
- Filling missing features
- Clustering
- Link prediction
  - Predict future new edges in a graph
- Community detection
  - Cluster nodes based on edge density in a graph
- Word embedding
  - Word2Vec is actually an MF

Quiz: 推薦系統之阿拉丁神燈Part 2

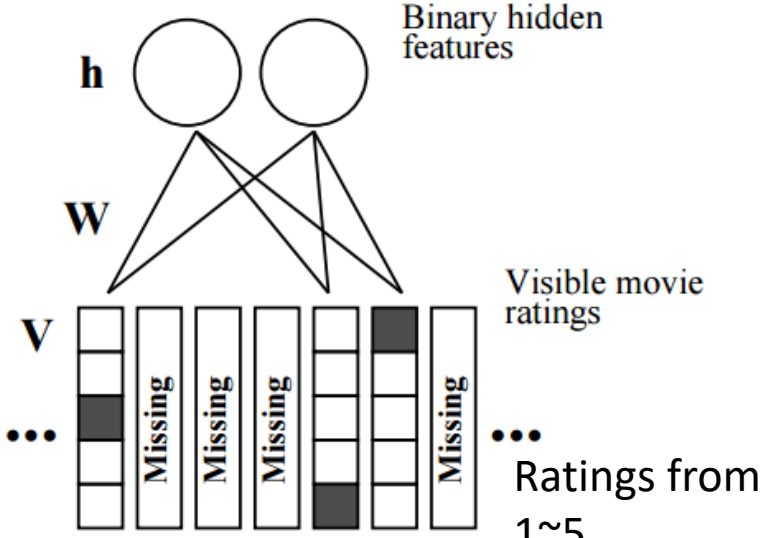
- 小胖撿到了一個神燈，他摩擦了神燈，神燈巨人答應他一個願望，小胖說：我想要蔡10陪我共進晚餐（這次不要找13了）。神燈巨人說沒問題，繃的一聲，就把蔡10，侯佩程，昆0都變了出來。
- 請問，神燈巨人內部是執行哪種推薦系統？

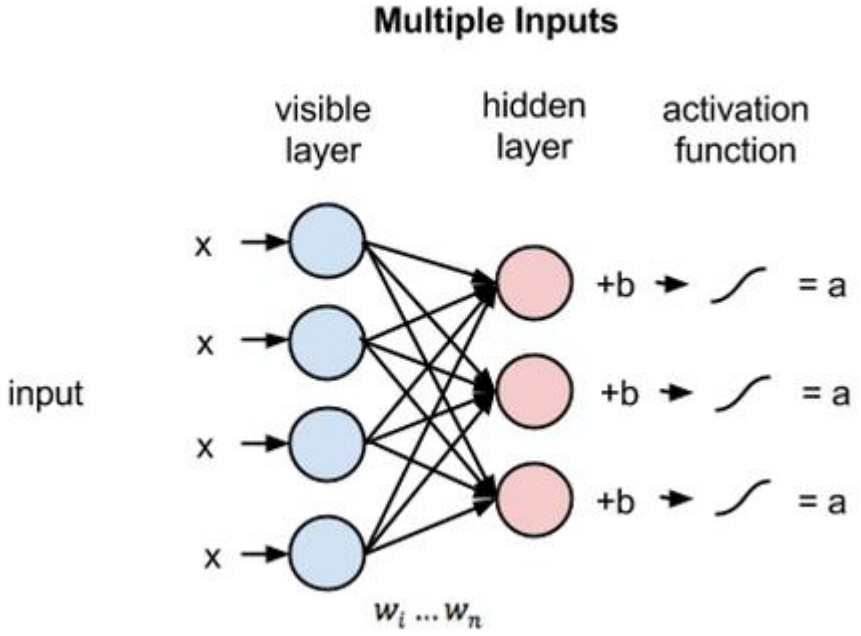
Types of Recommender Systems

- Content-based Recommendation (CBR)
- Collaborative Filtering
- Other solutions
  - Restricted Boltzmann Machines
  - Clustering-based recommendation
  - Association rule based recommendation
  - Random-walk based recommendation
- Advanced Issues

Restricted Boltzmann Machines For Collaborative Filtering

- RBM is an MRF of 2 layers
  - One RBM each user, but share the link weights
- Learned by Contrastive divergence
- DBM → Deep Boltzmann Machine





Restricted Boltzmann Machines for Collaborative Filtering, ICML 07

Clustering-based Recommendation

- Step 1: Clustering users into groups
- Step 2: finding the highest rated items in each group as the recommended items for this group
- Notes: it is more efficient but less personalized

	I1	I2	I3	I4	I5
U1	**4**			2	
U2	**4**				3
U3		**3**	2		2
U4		**4**	3		
U5		**3**			2

Association-rule based recommendation

- Finding association rules from global data
  - E.g. (item1, item2) appears together frequently
  - User 1 has purchased item1 → recommend item 2
- Pros: easy to implement, quick prediction
- Cons: it is not very personalized
- Associate rule work well for retail store recommendation

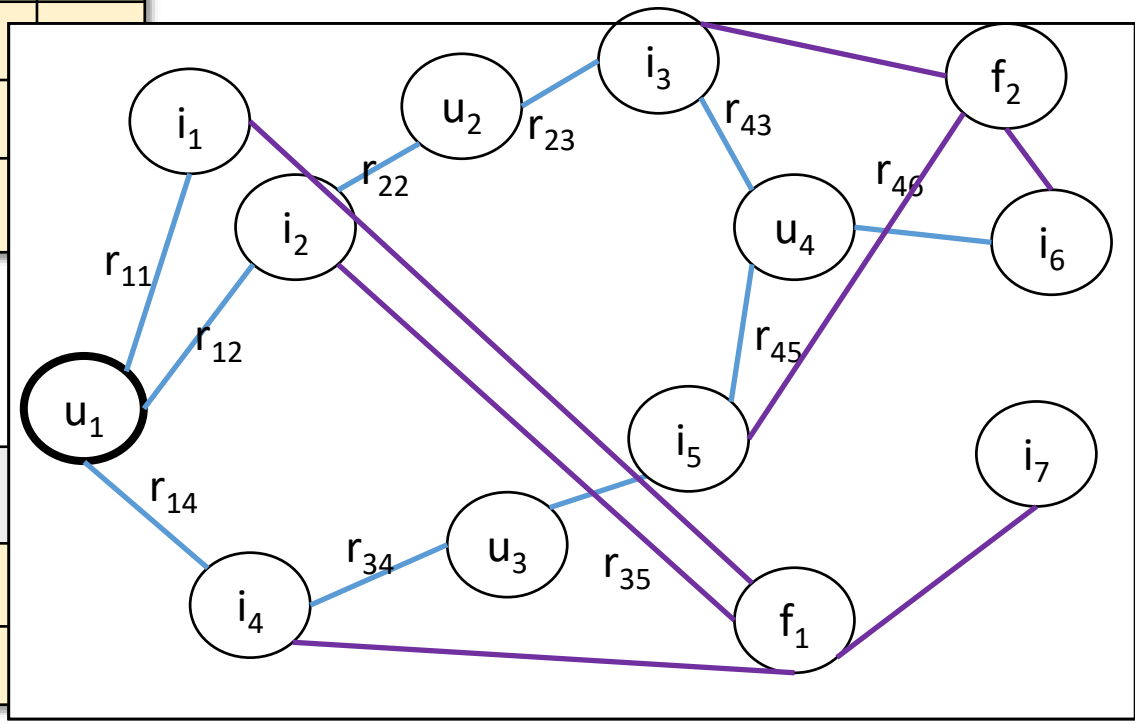
Graph-Based Model

- Build network from rating database

•	i_1	i_2	i_3	i_4	i_5	i_6	i_7
u_1	r_{11}	r_{12}		r_{14}			
u_2		r_{22}	r_{23}				
u_3				r_{34}	r_{35}		
u_4			r_{43}		r_{45}	r_{46}	

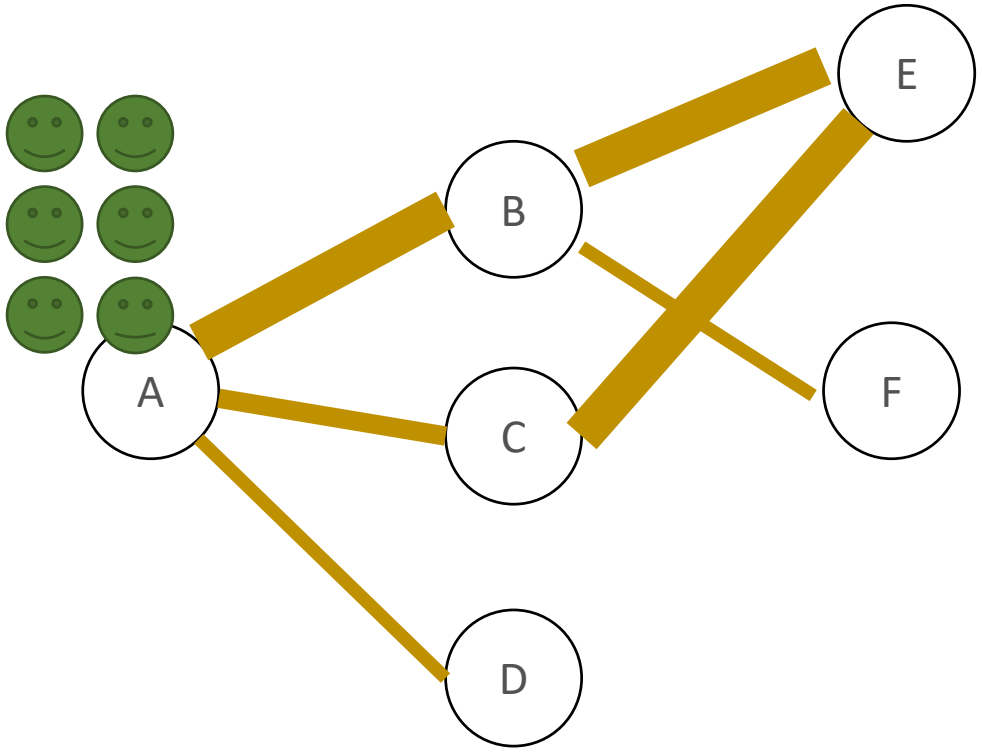
- Extra knowledge

	i_1	i_2	i_3	i_4	i_5	i_6
f_1	v	v		v		
f_2			v		v	v



Random Walk on Graph

- Random walk with restart
  - Random surf in the network from a source user.
  - Recommend items that have higher chance to be reached



Limitations on Collaborative filtering

1. Cannot consider features of items and users
   - Solution: factorization machine
2. Cannot consider cold-start situation
   - Solution: transfer recommendation

Demo: A joke recommendation system using CM models

- http://eigentaste.berkeley.edu/index.html

Evaluating Recommendation Systems

- Accuracy of predictions
  - How close predicted ratings are to the true ratings
- Relevancy of recommendations
  - Whether users find the recommended items relevant to their interests
- Ranking of recommendations
  - Ranking products based on their levels of interestingness to the user

Accuracy of Predictions

- Mean absolute error (MAE)
  - $MAE = \frac{\sum_{ij}|\hat{r}_{ij} - r_{ij}|}{n}$
  - $\hat{r}_{ij}$: Predicted rating of user i and item j
  - r_{ij}: True rating
- Normalized mean absolute error (NMAE)
  - $NMAE = \frac{MAE}{r_{max} - r_{min}}$
- Root mean squared error (RMSE)
  - $RMSE = \sqrt{\frac{1}{n}\sum_{ij}\left(\hat{r}_{ij} - r_{ij}\right)^2}$
  - Error contributes more to the RMSE value

Example of Accuracy

- $MAE = \frac{|1-3|+|2-5|+|3-3|+|4-2|+|4-1|}{5} = 2$
- $NMAE = \frac{MAE}{5-1} = 0.5$
- $RMSE = \sqrt{\frac{(1-3)^2+(2-5)^2+(3-3)^2+(4-2)^2+(4-1)^2}{5}} = 2.28$

Item	Predicted Rating	True Rating
1	1	3
2	2	5
3	3	3
4	4	2
5	4	1

Relevancy of Recommendations

- Precision
  - $P = \frac{N_{rs}}{N_s}$
- Recall
  - $R = \frac{N_{rs}}{N_r}$
- F-measure (F_1 score)
  - $F = \frac{2PR}{P+R}$

		Recommended Items		
		Selected	Not Selected	Total
Relevancy	Relevant	N_{rs}	N_{rn}	N_r
	Irrelevant	N_{is}	N_{in}	N_i
	Total	N_s	N_n	N

Example of Relevancy

- $P = \frac{9}{12} = 0.75$
- $R = \frac{9}{24} = 0.375$
- $F = \frac{2 \times 0.75 \times 0.375}{0.75 + 0.375} = 0.5$

	Selected	Not Selected	Total
Relevant	9	15	24
Irrelevant	3	13	16
Total	12	28	40

Ranking of Recommendations

- Discounted cumulative gain (DCG)
  - $DCG = \frac{1}{|U|}\sum_{u_i \in U}\sum_{j=1}^{L}\frac{\hat{R}_{i,j}}{\max\{1,\log_2 j\}}$
  - There are L items in the ranked list
  - U: User set
- Spearman's rank correlation: For n items
  - $\rho = 1 - \frac{6\sum_{i=1}^{n}(x_i - y_i)^2}{n^3 - n}$
  - $1 \leq x_i \leq n$: Predicted rank of item i
  - $1 \leq y_i \leq n$: True rank of item i
- Kendall's tau: For all $\binom{n}{2}$ pairs of items (i, j)
  - $\tau = \frac{c-d}{\binom{n}{2}}$, in range $[-1,1]$
  - There are c concordant pairs
    - $x_i > x_j, y_i > y_j$ or $x_i < x_j, y_i < y_j$
  - There are d discordant pairs
    - $x_i > x_j, y_i < y_j$ or $x_i < x_j, y_i > y_j$

Example of Ranking

- Kendall's Tau
  - $\tau = \frac{4-2}{6} = 0.33$
    - (i_1, i_2): concordant
    - (i_1, i_3): concordant
    - (i_1, i_4): concordant
    - (i_2, i_3): discordant
    - (i_2, i_4): discordant
    - (i_3, i_4): concordant

Item	Predicted Rank	True Rank
i_1	1	1
i_2	2	4
i_3	3	2
i_4	4	3

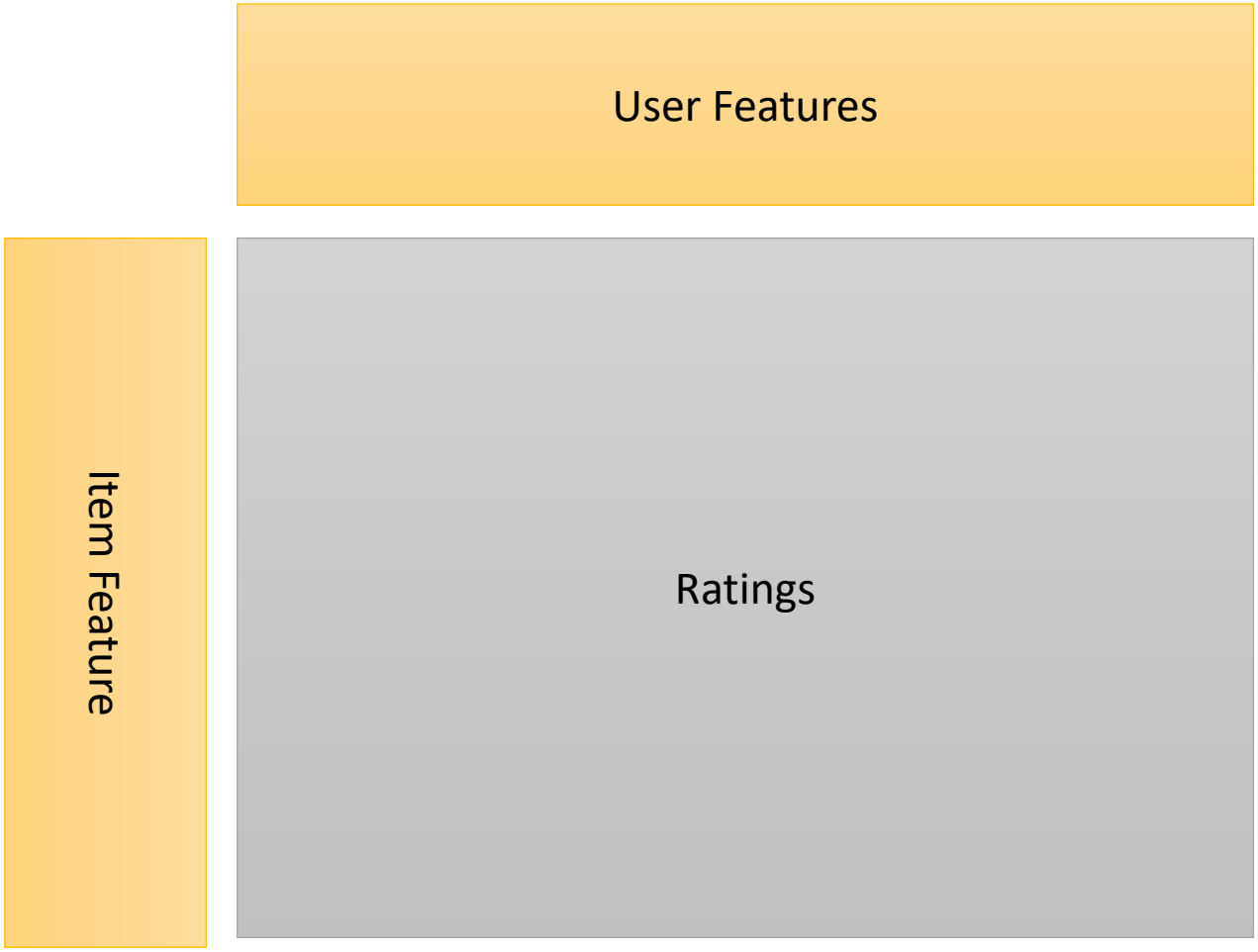
- Spearman's rank correlation

$$-\rho = 1 - \frac{6\left((1-1)^2+(2-4)^2+(3-2)^2+(4-3)^2\right)}{4^3-4} = 0.4$$

Types of Recommender Systems

- Content-based Recommendation (CBR)
- Collaborative Filtering
- Other solutions
  - Restricted Boltzmann Machines
  - Clustering-based recommendation
  - Association rule based recommendation
  - Random-walk based recommendation
- Advanced Issues

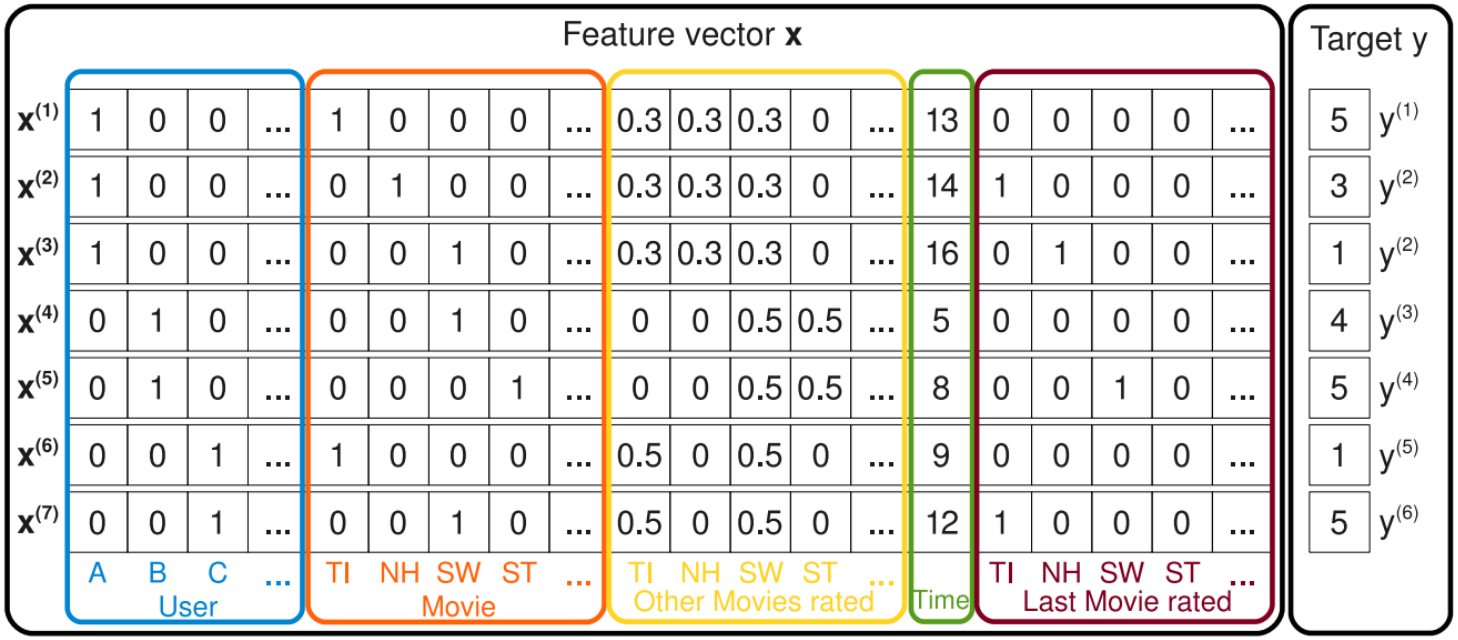
Advanced issue 1: Recommendation with ratings and features



Factorization Machine (FM)

- Matrix Factorization (MF) $Y \approx U^T V$
- Factorization Machine $Y \approx f(x)$, x is highly sparse

  Model: $$\hat{y}(x) := w_0 + \sum_{i=1}^{n} w_i x_i + \sum_{i=1}^{n} \sum_{j=i+1}^{n} \langle v_i, v_j \rangle x_i x_j$$



Steffen Rendle "factorization machines" 2010

Advance Issue 2: Handling Cold Start

- What happens if we don't have rating/review information for some items or users
- Solution: transfer information from other domains
  - Use book review data to improve a movie recommender system
  - Use ratings from one sources (e.g. Yhaoo) to improve the recommender of another (e.g. Pchome).

Why Transfer Recommendation?

- Nowadays, users can
  - provide feedbacks for items of different types • e.g., in Amazon we can rate books, DVDs, …
  - express their opinions on different social media and different providers (Facebook, Twitter, Amazon)
- Transfer recommendation tries to utilize information from source domain to Improve the quality of an RS
  - Solving cold-start problem (very few ratings for some users or some items)
  - Improving accuracy

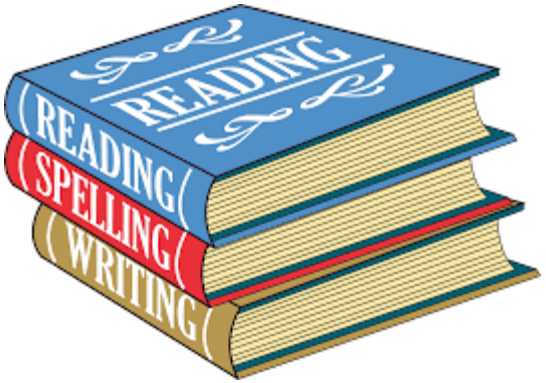
Transfer between domains

- Source domain: contains knowledge to transfer to target, usually more data.
- Target domain: needs help from the source domain, usually contains cold start users or items.
- Domains differ because of
  - different types of items
    - Movies vs. Books
  - different types of users
    - American vs. Taiwanese

Source	Target
Movies	Music
Pop music	Jazz music
Taiwan movie	US movies

Case Study: Why Transfer Recommendation is important for New Business

You are the CEO of an online textbook store Ambzon.com which used to sell only English books in USA.



You want to expand the business across the world to sell some Chinese textbooks in Taiwan.

- You need a recommender system for such purpose.
- Unfortunately, you only have few user-item rating records for the Chinese textbooks rated by Taiwanese.

However, you have much more ratings on the English textbooks rated by the American people

User\ Book	B1	B2	B3	B4	B5
U1	1	4	3	3	
U2			2		1
U3	3	5			2
U4			4	3	6
U5		2	2	4	
U6	4	6			

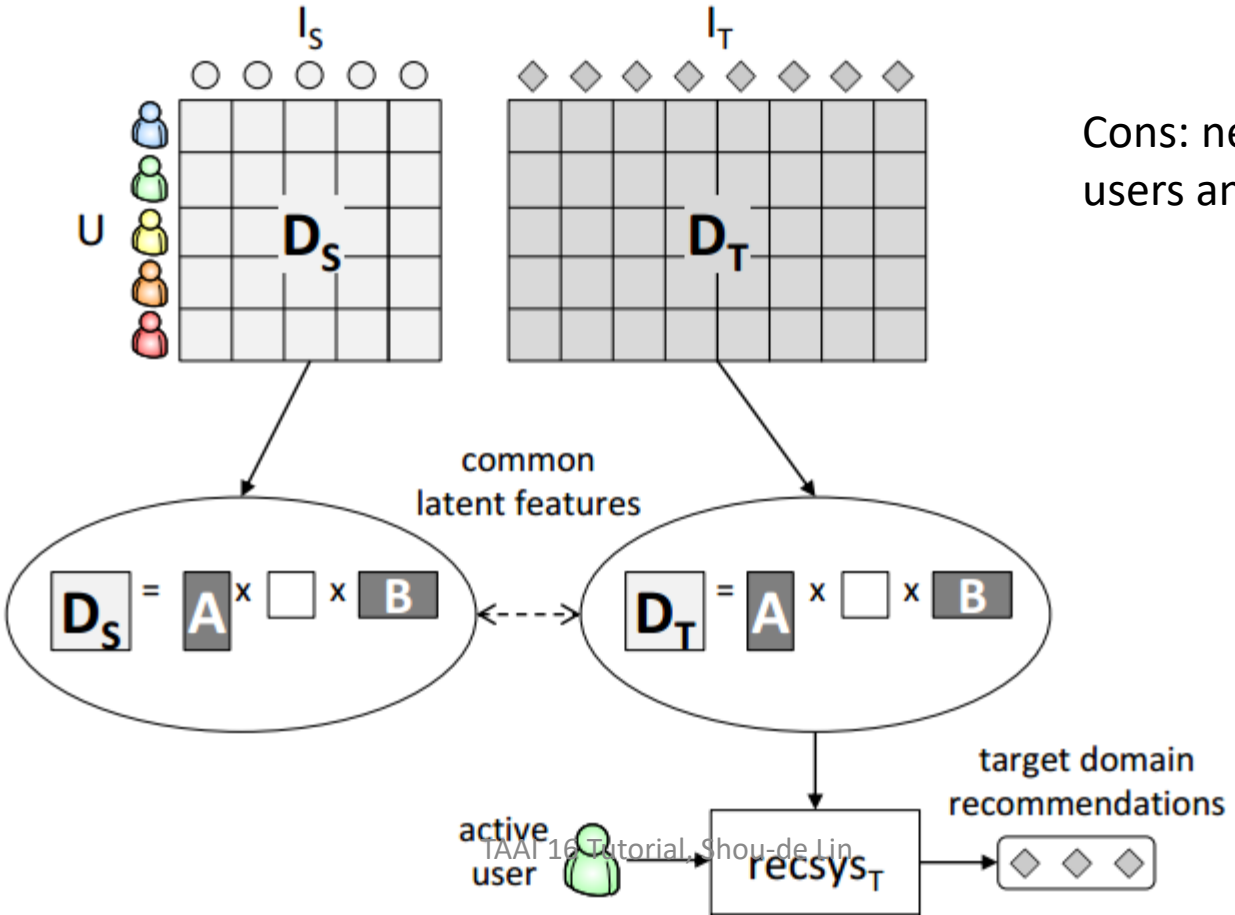
Challenge: You will have to leverage the ratings from the English textbooks to enhance the recommendation on Chinese textbooks?

How can Transfer be Done?

- Feature sharing
  - user demographic features in both domains
  - item attributes in both domains
- Linking users and items
  - Finding overlapping users and items across domains
  - Connecting users (e.g. friendship) and items (e.g. content) across domains

Sharing Latent Features of users/items

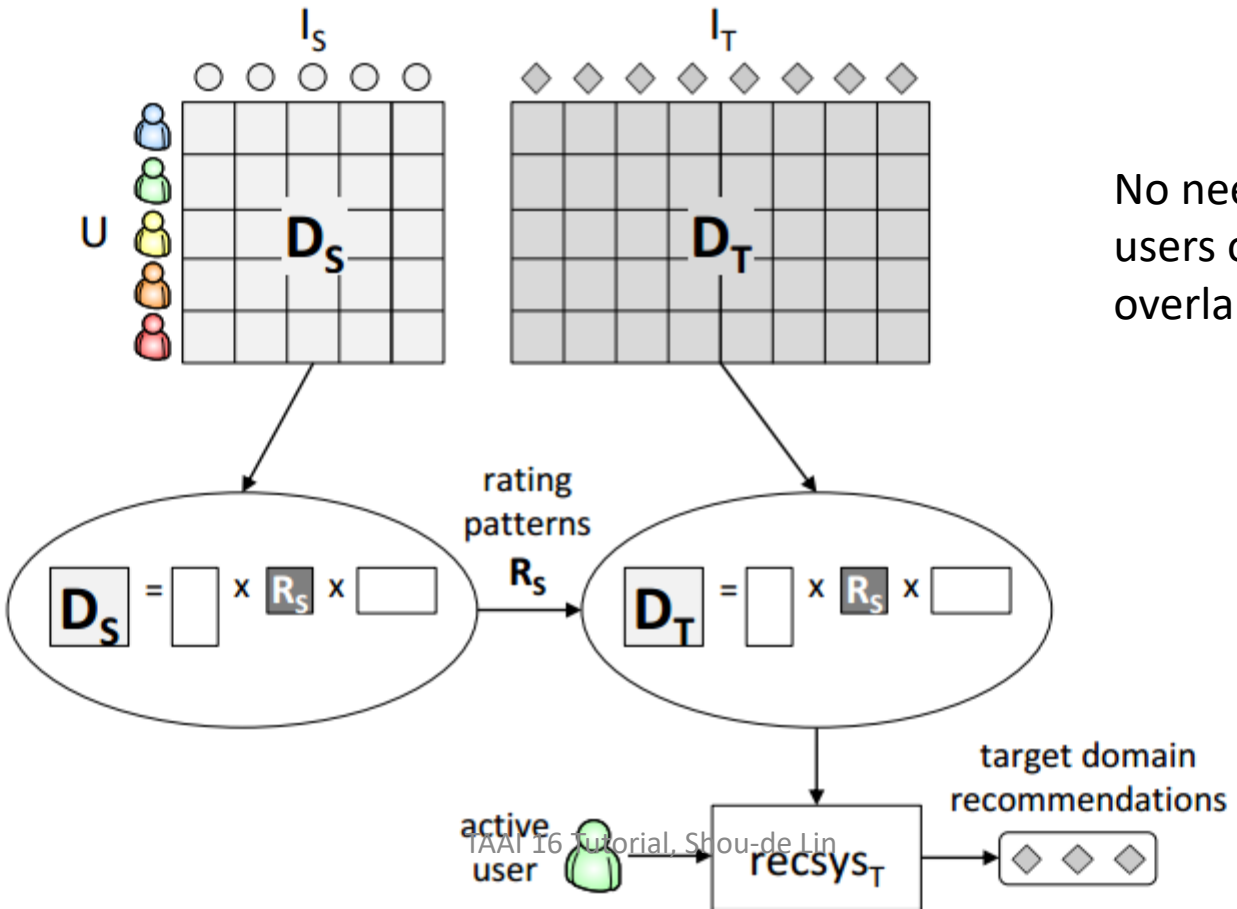
- source and target domains are related by means of shared latent features



Cons: need overlap of users and/or items

Transferring rating patterns

- rating patterns are transferred between domains

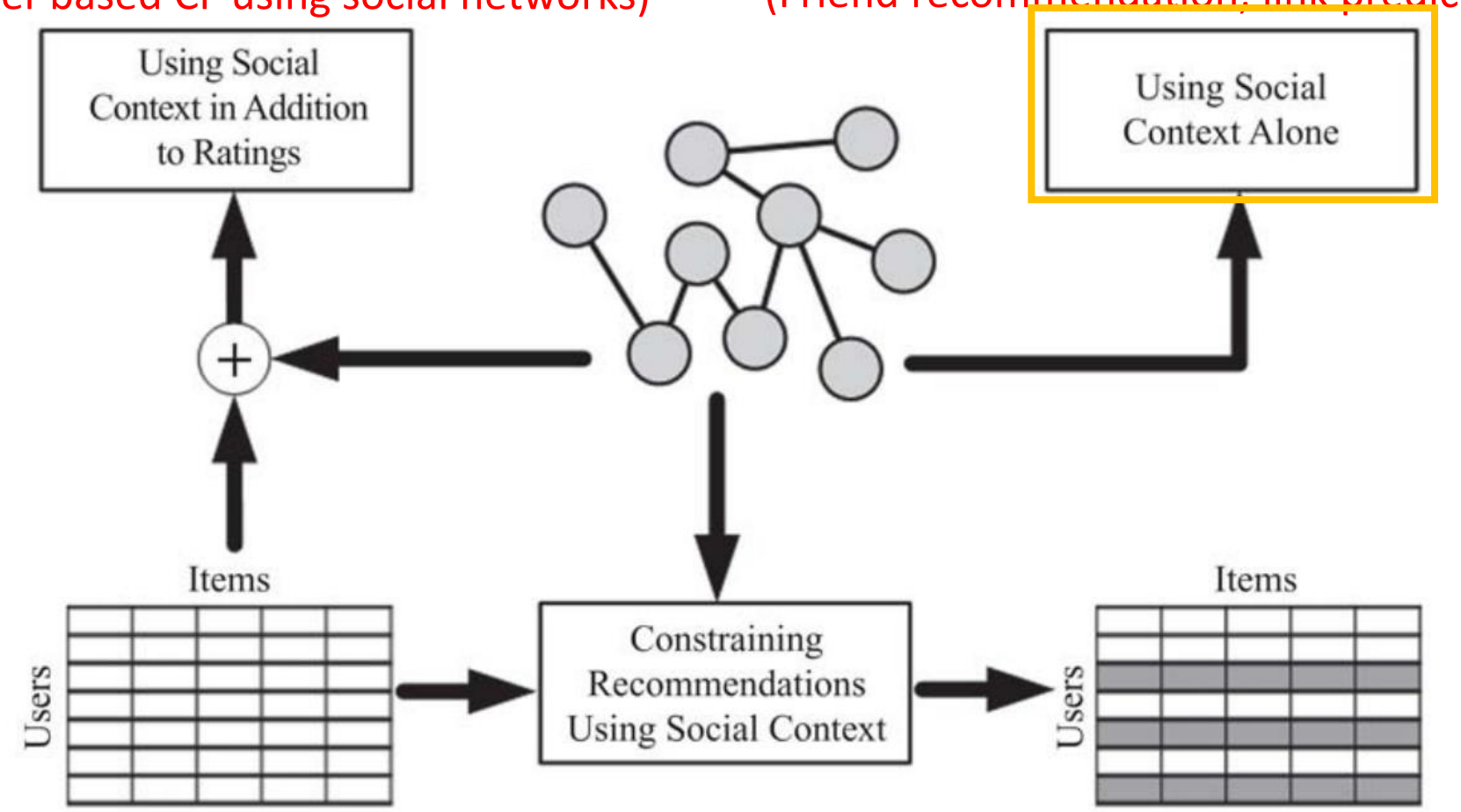


No need to assume users or items have overlap

Advanced Issues 3: Using Social information in Recommendation

(Model-based CF using social networks)

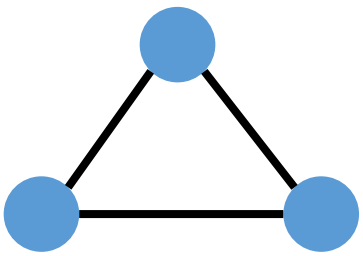
(Friend recommendation, link prediction)



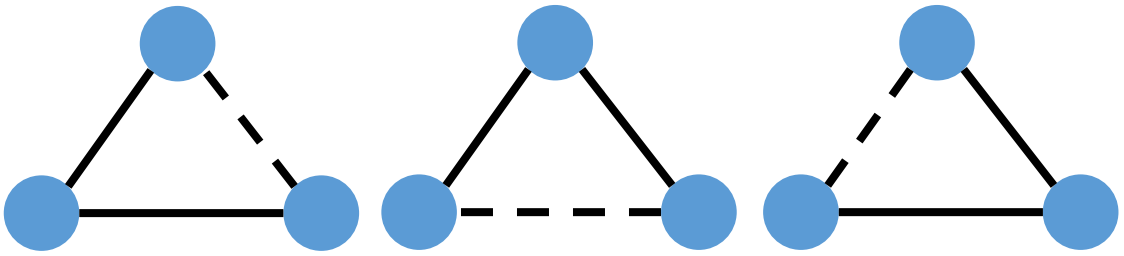
(Memory-based CF using social networks)

Social Context Alone

- Friend recommendation in social networks
- Methods
  - Link prediction
  - Network structure e.g. triad
    - Two friends of an individual are often friends
    - Friend recommendation: Open triad (missing one edge)

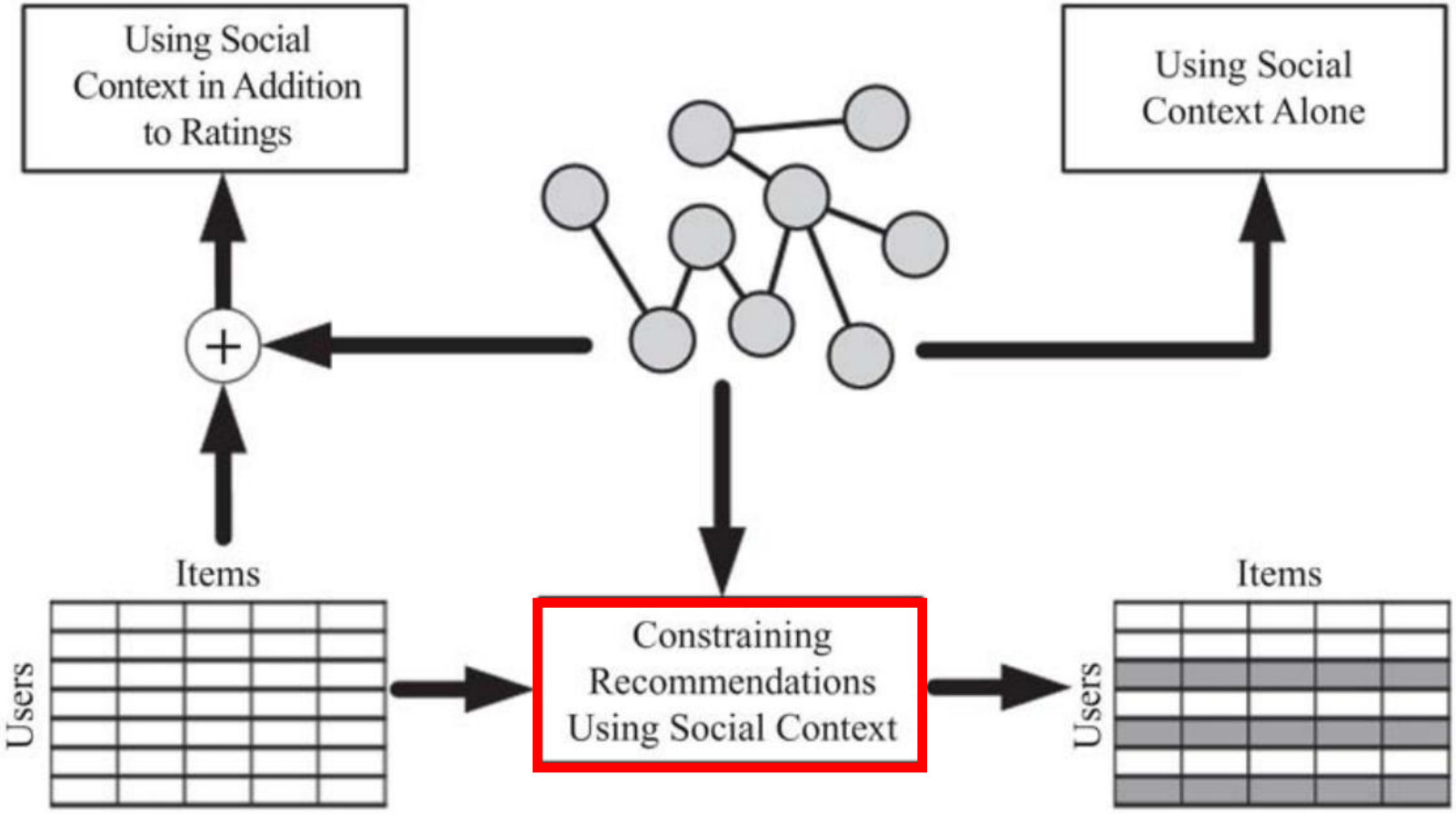


Triad

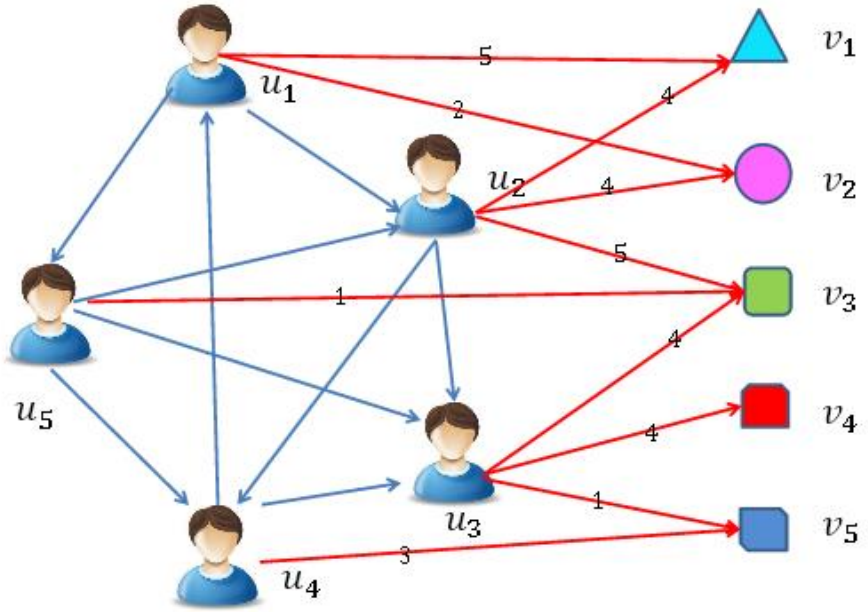


Open triad

Recommendation Using Social Context



Social Rec V.S. Collaborative Filtering



Social recommendation systems

Traditional recommendation systems

Items

Users	v_1	v_2	v_3	v_4	v_5
u_1	5	?	2	?	?
u_2	4	4	5	?	?
u_3	?	?	4	4	1
u_4	?	?	?	?	3
u_5	?	?	1	?	?

Rating matrix R

(b) Traditional Recommender Systems

	v_1	v_2	v_3	v_4	v_5
u_1	5	?	2	?	?
u_2	4	4	5	?	?
u_3	?	?	4	4	1
u_4	?	?	?	?	3
u_5	?	?	1	?	?

Rating matrix R

	u_1	u_2	u_3	u_4	u_5
u_1	0	1	0	0	1
u_2	0	0	1	1	0
u_3	0	0	0	0	0
u_4	1	0	1	0	0
u_5	0	1	1	1	0

Social matrix T

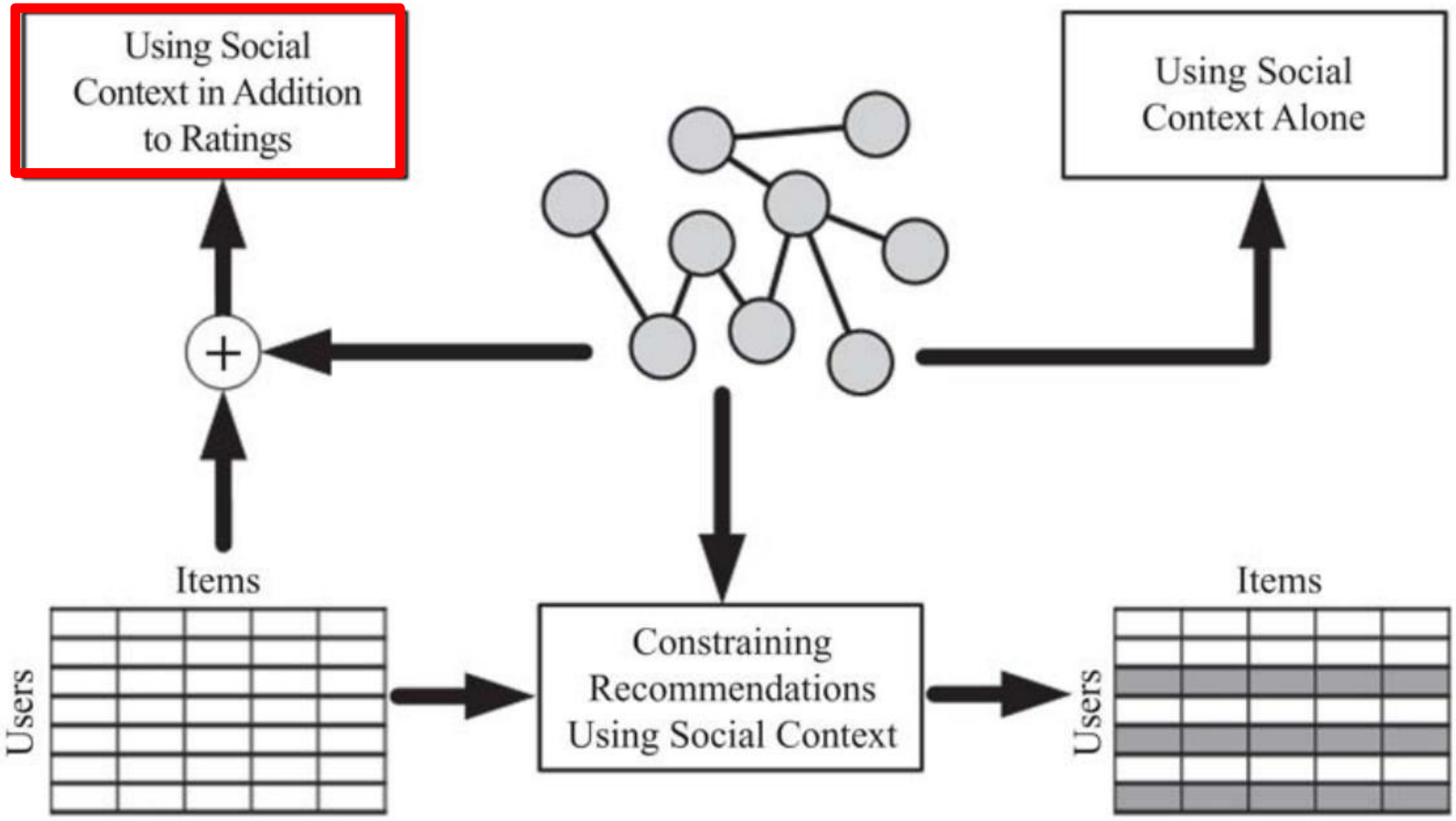
(c) Social Recommender Systems

Memory-Based Social Recommendation

- Especially for user-based CF
- Constraining the set of correlated users
  - Traditional: N(u) obtained from rating matrix R
  - Social: N^+ (u) obtained from both the rating matrix R and social matrix T
- The prediction model (aggregation of the ratings from the correlated users) remains the same

$$r_{u,i} = \overline{r}_u + \frac{\sum_{v \in N^+(u)} sim(u,v)(r_{v,i} - \overline{r}_v)}{\sum_{v \in N^+(u)} sim(u,v)}$$

Recommendation Using Social Context



Social Regularization Model

- Add an additional regularization term to MF optimization function:

$$\min_{P,Q} \left\{ \begin{array}{c} \sum_{r_{ix}} (r_{ix} - q_i p_x)^2 \\ +\lambda \left[\sum_i \|q\|^2 + \sum_x \|p_x\|^2 \right] \\ +\beta \sum_{x,y} w_{xy} \|p_x - p_y\|^2 \end{array} \right\}$$

  - w_{xy}: strength of the connection between x and y
  - $\|p_x - p_y\|$: the difference between the latent preferences of the users.

H. Ma et al. Recommender systems with social regularization, WSDM2011

Advanced Issue 4: Implicit Recommendation

- Sometimes it is hard to obtain user ratings toward items.
- However, it is easier to obtain some implicit information from users to items, such as
  - Browsing
  - Purchasing
  - Listen/watch

- **The rating matrix contains only '1' and '?'**
- **Regular MF would predict every '?' to 1 to minimize the squared error**

Implicit ratings	Item1	Item2	Item3	Item4
User1	?	1	?	?
User2	1	?	?	1
User3	?	1	?	1
User4	1	?	1	?

Solution: Bayesian Personalized Ranking –MF (BPR-MF)➔ minimize pairwise ranking

Advanced Issue 5: Location-based Recommendation

- Learning Travel Recommendations from User-generated GPS Traces. ACM TIST 2011.
- Measuring and Recommending Time-Sensitive Routes from Location-based Data. ACM TIST 2014.
- *New Venue Recommendation in Location-Based Social Networks*.
  IEEE SocialCom 2012.
- *Exploiting Geographical Influence for Collaborative Point-of-Interest Recommendation*. ACM SIGIR 2011.
- *Time-aware Point-of-Interest Recommendation*. ACM SIGIR 2013.

Advanced Issue 6: Explaining the Recommendation Outputs

- With reasonable explanation, recommendation can be more powerful
  - Recommendation → Persuasion
- Goal: how can the system automatically generates human-understandable explanations for the recommendation.

Explanation in recommender systems, AIR 2005

Explaining collaborative filtering recommendations CSCW 2000

Quiz:推薦系統之阿拉丁神燈part 3

- 小胖跟蔡10自從上次共進晚餐後互有好感，於是FB狀態改成「穩定交往中」。有天，蔡10跟小胖借了神燈來許願，她說：「我平常工作很忙，沒有什麼好朋友，你可以推薦一個朋友給我嗎？」
- 神燈巨人說：沒問題。於是繃的一聲，一個女孩變了出來。小胖看到她臉色大變，原來出現的是他的前女友小文。神燈巨人說：「我推薦小文給蔡10，因為妳們兩個有共同好友！！！！！」
- 請問，神燈巨人是執行何種推薦系統？

Library for Recommender Systems

LIBMF: A Matrix-factorization Library for Recommender Systems

URL: https://www.csie.ntu.edu.tw/~cjlin/libmf/

Language: C++

Focus: solvers for Matrix-factorization models

R interface: package “recosystem”

https://cran.r-project.org/web/packages/recosystem/index.html

Features:

- solvers for real-valued MF, binary MF, and one-class MF
- parallel computation in a multi-core machine
- less than 20 minutes to converge on a data set of 1.7B ratings
- supporting disk-level training, which largely reduces the memory usage

MyMediaLite - a recommender system algorithm library

URL: http://mymedialite.net/

Language: C#

Focus: rating prediction and item prediction from positive-only feedback

Alogrithms: kNN, BiasedMF, SVD++...

Features:

Measure: MAE, RMSE, CBD, AUC, prec@N, MAP, and NDCG

LibRec: A Java Library for Recommender Systems

URL: http://www.librec.net/index.html

Language: Java

Focus: algorithms for **rating prediction** and **item ranking**

Algorithms: kNN, PMF, NMF, BiasedMF, SVD++, BPR, LDA…more at: http://www.librec.net/tutorial.html

Features:

Faster than MyMediaLite

Collection of Datasets: MovieLens 1M, Epinions, Flixster...

http://www.librec.net/datasets.html

mrec: recommender systems library

URL: http://mendeley.github.io/mrec/

Language: Python

Focus: item similarity and other methods for implicit feedback

Algorithms: item similarity methods, MF, weighted MF for implicit feedback

Features:

- train models and make recommendations in parallel using IPython
- utilities to prepare datasets and compute quality metrics

SUGGEST: Top-N recommendation engine

Python interface: “pysuggest”

https://pypi.python.org/pypi/pysuggest

Focus: collaborative filtering-based top-N recommendation algorithms (user-based and item-based)

Algorithms: user-based or item-based collaborative filtering based on various similarity measures

Features:

low latency: compute top-10 recommendations in less that 5us

Conclusion

- Recommendation is arguably the most successful AI/ML solutions till now.
  - It is not just for customer-product
  - Matching users and users
  - Matching users and services
  - Matching users and locations
  - …
- The basic skills and tools are mature, but the advanced issues are not fully solved.

References (1/3)

- SVD++
  - Koren, Yehuda. "Factorization meets the neighborhood: a multifaceted collaborative filtering model." *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2008.
- TF
  - Karatzoglou, Alexandros, et al. "Multiverse recommendation: n-dimensional tensor factorization for context-aware collaborative filtering." *Proceedings of the fourth ACM conference on Recommender systems*. ACM, 2010.
- FM
  - Rendle, Steffen, et al. "Fast context-aware recommendations with factorization machines." *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*. ACM, 2011.
- BPR-MF
  - Rendle, Steffen, et al. "BPR: Bayesian personalized ranking from implicit feedback." *Proceedings of the twenty-fifth conference on uncertainty in artificial intelligence*. AUAI Press, 2009.
- BPMF
  - Salakhutdinov, Ruslan, and Andriy Mnih. "Bayesian probabilistic matrix factorization using Markov chain Monte Carlo." *Proceedings of the 25th international conference on Machine learning*. ACM, 2008.
- PF
  - Gopalan, Prem, Jake M. Hofman, and David M. Blei. "Scalable recommendation with poisson factorization." *arXiv preprint arXiv:1311.1704* (2013).

References (2/3)

- Link prediction
  - Menon, Aditya Krishna, and Charles Elkan. "Link prediction via matrix factorization." *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer Berlin Heidelberg, 2011.
- Community detection
  - Yang, Jaewon, and Jure Leskovec. "Overlapping community detection at scale: a nonnegative matrix factorization approach." *Proceedings of the sixth ACM international conference on Web search and data mining*. ACM, 2013.
  - Psorakis, Ioannis, et al. "Overlapping community detection using bayesian non-negative matrix factorization." *Physical Review E* 83.6 (2011): 066114.
- Clustering
  - Ding, Chris HQ, Xiaofeng He, and Horst D. Simon. "On the Equivalence of Nonnegative Matrix Factorization and Spectral Clustering." *SDM*. Vol. 5. 2005.
  - Kuang, Da, Sangwoon Yun, and Haesun Park. "SymNMF: nonnegative low-rank approximation of a similarity matrix for graph clustering." *Journal of Global Optimization* 62.3 (2015): 545-574.
- Word embedding
  - Levy, Omer, and Yoav Goldberg. "Neural word embedding as implicit matrix factorization." *Advances in neural information processing systems*. 2014.
  - Pennington, Jeffrey, Richard Socher, and Christopher D. Manning. "Glove: Global Vectors for Word Representation." *EMNLP*. Vol. 14. 2014.

References (3/3)

- The Matrix Cookbook
  - http://www2.imm.dtu.dk/pubdb/views/edoc_download.php/3274/pdf/imm3274.pdf
  - Collection of matrix facts, including derivatives and expected values