

Unsupervised learning: Movie Recommendation

ผศ. ดร. เก็จแก้ว ธเนศวร

kejkaew.tha@mail.kmutt.ac.th

ครั้งที่	วันที่	หัวข้อ	การบ้าน	ตรวจงาน
1	18/1/2567	Introduction to Artificial Intelligence + Python 0	Assignment 1 (KT,KN)	1
2	25/1/2567	Python Programming 1	Assignment 2 (KT,KN)	1,2
3	1/2/2567	Python Programming 2	Assignment 3 (KT,KN)	1,2,3
4	8/2/2567	Data Structure in Python	Assignment 4 (KT, KN)	2,3,4
5	15/2/2567	Search 1: 8 Puzzle game	Assignment 5 (KT)	3,4,5
6	29/2/2567	Search 2: 8 Puzzle game	Assignment 6 (KT,KN)	4,5,6
7	7/3/2567	Adversarial search: Tic-Tac-Toe	Assignment 7 (KT,KN)	5,6,7
	14/3/2567	สอบกลางภาค		
8	21/3/2567	Machine learning: Dead or Alive	Assignment 8 (KT, KN)	6,7,8
9	28/3/2567	Machine learning: Cars	Assignment 9 (KT,KN)	7,8,9
10	4/4/2567	Unsupervised learning: Who are your customers?	Assignment 10 (KT,PP)	8,9,10
	14/4/2567	(วันหยุด)	ส่งงานเดี่ยว	
11	18/4/2567	Unsupervised learning: Movie Recommendation	Assignment 11 (KT, PP)	9,10,11
	25/4/2567	(ไม่อยู่)		
12	2/5/2567	Computer vision: Blackpink in your image?	Assignment 12 (KT, PP)	10,11,12
13	9/5/2567	ส่งงาน online	KT,PP	11,12
14	16/5/2567	ส่งโปรเจค	PP	12
15	23/5/2567	สอบปลายภาค	ส่งงานกลุ่ม	

Last Week Topics

- Unsupervised learning: Similarity
- Clustering
- Assignment 10

Today Topics

- Recommendation system
 - Content-based systems
 - Collaborative filtering systems
- Assignment 11
- Project

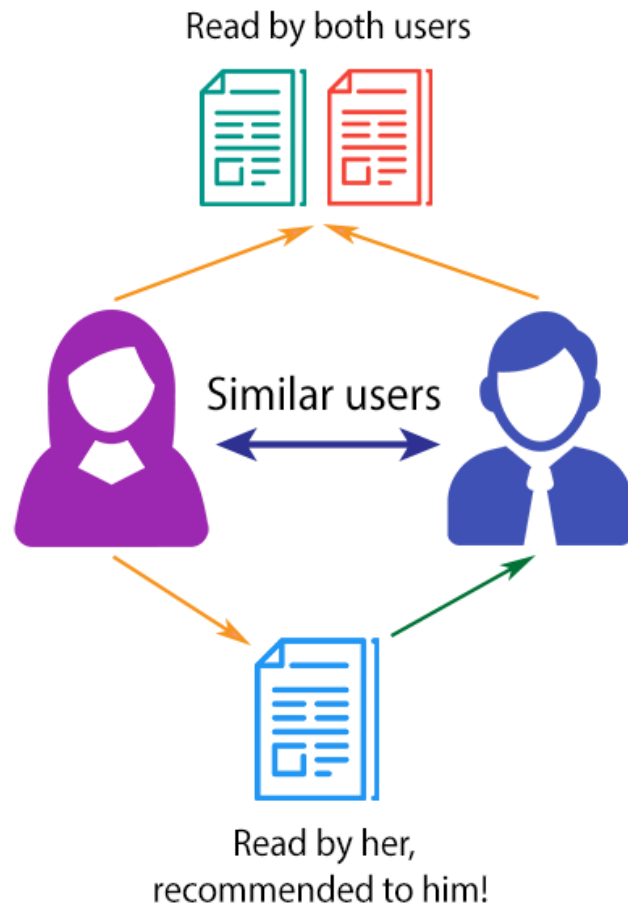
Co-occurrence analysis

Recommender Engine

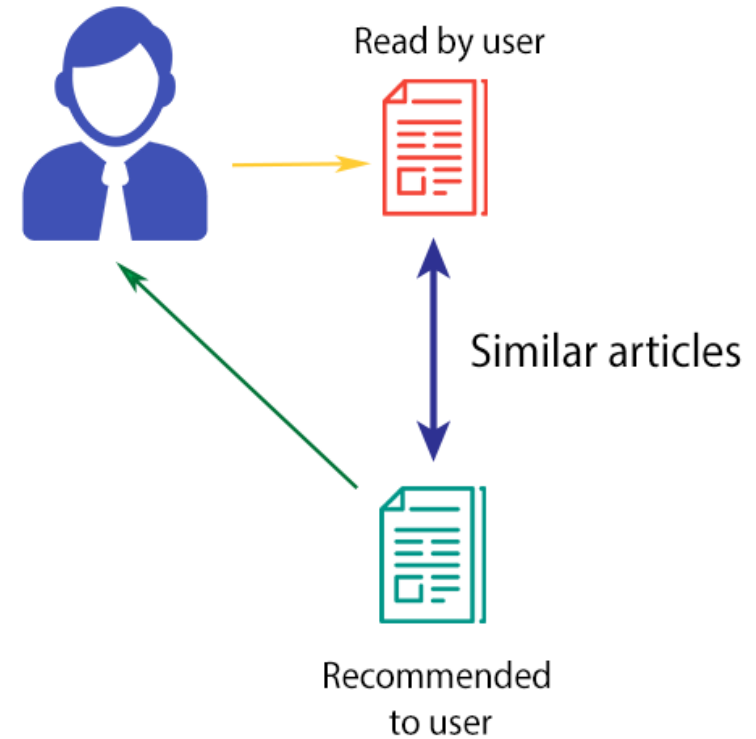
What is a recommender?

- A system that provides recommendation
- We can classify these systems into two broad groups.
 - **Content-based systems** examine properties of the items recommended, e.g. Netflix
 - **Collaborative filtering systems** recommend items based on similarity measures between users and/or items. The items recommended to a user are those preferred by similar users.

COLLABORATIVE FILTERING



CONTENT-BASED FILTERING



Applications of recommender systems (Example)

1. **Product Recommendations:** Perhaps the most important use of recommendation systems is at on-line retailers.
2. **Movie Recommendations:** Netflix offers its customers recommendations of movies they might like.
3. **News Articles:** News services have attempted to identify articles of interest to readers, based on the articles that they have read in the past.

Utility Matrix

Example 9.1 : In Fig. 9.1 we see an example utility matrix, representing users' ratings of movies on a 1–5 scale, with 5 the highest rating. Blanks represent the situation where the user has not rated the movie. The movie names are HP1, HP2, and HP3 for *Harry Potter* I, II, and III, TW for *Twilight*, and SW1, SW2, and SW3 for *Star Wars* episodes 1, 2, and 3. The users are represented by capital letters *A* through *D*.

	HP1	HP2	HP3	TW	SW1	SW2	SW3
<i>A</i>	4			5	1		
<i>B</i>	5	5	4				
<i>C</i>				2	4	5	
<i>D</i>		3					3

Content-based or Collaborative

- Movie recommendation example:
 1. Predict if a user likes an item based on the item descriptions (movie genres). (content-bases)
 2. Assume that users like similar items, and retrieve movies that are closest in similarity to a user's profile, which represents a user's preference for an item's feature. (collaborative)

Content-based filtering

Content-based recommendation: profiles

- In a content-based system, we must construct for each item a profile, which is a record or collection of records representing important characteristics of that item.
- For examples,
 - **The set of actors** of the movie. Some viewers prefer movies with their favorite actors.
 - **The director**. Some viewers have a preference for the work of certain directors.
 - **The year** in which the movie was made. Some viewers prefer old movies; others watch only the latest releases.
 - **The genre or general type of movie**. Some viewers like only comedies, others dramas or romances.

Content-based filtering

Good

- It does not require a lot of user data.
- It just needs item data and you are able to start giving recommendations to users.
- It does not depend on lots of user data, so it is possible to give recommendations to even your first customer.

Bad

- Your item data needs to be well distributed.
- It won't be effective to have a content-based recommender if 80% of your movies are action movies.
- Complements are more likely discovered through collaborative techniques.

Cosine Similarity

Cosine similarity measures the similarity between two vectors of an inner product space.

คือ การหาความคล้ายด้วยองศา

สิ่งที่ต้องรู้ คือ การ Dot product

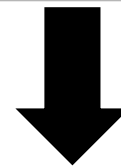
$$\mathbf{u} \cdot \mathbf{v} = [u_1 \ u_2 \ \dots \ u_n] \cdot \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix} = u_1 v_1 + u_2 v_2 + \dots + u_n v_n = \sum_{i=1}^n u_i v_i$$

Cosine Similarity

$$\text{similarity} = \cos(\theta) = \frac{\mathbf{u} \cdot \mathbf{v}}{\|\mathbf{u}\| \|\mathbf{v}\|} = \frac{\sum_{i=1}^n u_i v_i}{\sqrt{\sum_{i=1}^n u_i^2} \sqrt{\sum_{i=1}^n v_i^2}}$$

Cosine similarity

	Movie_id	Word1 – 'action'	Word2 – 'animation'	Word3- 'horror'	...	Word5000 – 'guitar'
vector1	1001	5	3	9	..	10
vector2	1002	4	0	6	..	6
..
vector5000	6000	3	1	0	..	0



Cosine Similarity

$$\begin{matrix} & Movie_1 & Movie_2 & Movie_3 & \dots & Movie_n \\ \begin{matrix} Movie_1 \\ Movie_2 \\ Movie_3 \\ \vdots \\ Movie_n \end{matrix} & \begin{pmatrix} 1 & 0.158 & 0.138 & \dots & 0.056 \\ 0.158 & 1 & 0.367 & \dots & 0.056 \\ 0.138 & 0.367 & 1 & \dots & 0.049 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0.056 & 0.056 & 0.049 & \dots & 1 \end{pmatrix}
 \end{matrix}$$

TF-IDF

Term Frequency-inverse Document Frequency

เป็นเทคนิคที่แสดงถึงความสำคัญของคำๆ หนึ่งที่มีในแต่ละเอกสารโดยคิดจากเอกสารที่มีทั้งหมด

ซึ่งหมายถึงการนำ term frequency (จำนวนครั้งที่แต่ละ word id ปรากฏ ในแต่ละ text)หารด้วยจำนวน word ทั้งหมดใน text นั้น จากนั้นจึงนำมาคูณกับ log ของจำนวน document ทั้งหมด หารด้วย จำนวน document ที่แต่ละ word id นั้นปรากฏอยู่

$$w_{x,y} = \text{tf}_{x,y} \times \log \left(\frac{N}{\text{df}_x} \right)$$

TF-IDF

Term x within document y

$\text{tf}_{x,y}$ = frequency of x in y

df_x = number of documents containing x

N = total number of documents

Bag of Words

วิธีการทำ document classification อย่างง่าย โดยการนับความถี่ของคำในปรากฏ ในแต่ละ document

	about	bird	heard	is	the	word	you
About the bird, the bird, bird bird bird	1	5	0	0	2	0	0
You heard about the bird	1	1	1	0	1	0	1
The bird is the word	0	1	0	1	2	1	0

Movies



Vector-space representation

	Movie_id	Word1 – 'action'	Word2 – 'animation'	Word3- 'horror'	...	Word5000 – 'guitar'
vector1	1001	5	3	9	..	10
vector2	1002	4	0	6	..	6
..
vector5000	6000	3	1	0	..	0

Activity 1







ใน colab

Collaborative filtering




User-Based Collaborative Filtering

- The User-Based Collaborative Filtering approach groups users according to prior usage behavior or according to their preferences, and then recommends an item that a similar user in the same group viewed or liked.
- To put this in layman terms,
 - User 1 liked movie A, B and C
 - User 2 liked movie A and B
 - Then movie C might make a good recommendation to User 2.
- The User-Based Collaborative Filtering approach mimics how word-of-mouth recommendations work in real life.

User-Based Collaborative Filtering



	Item 1	Item 2	Item 3	Item 4	Item 5
User 1 	8	1	?	2	7
User 2 	2	?	5	7	5
User 3 	5	4	7	4	7
User 4 	7	1	7	3	8
User 5 	1	7	4	6	5
User 6 	8	3	8	3	7

Similarity between users

	Item 1	Item 2	Item 3	Item 4	Item 5
User 1 	8	1	?	2	7
User 2 	2	?	5	7	5
User 4 	7	1	7	3	8

- How similar are users 1 and 2?
- How similar are users 1 and 5?
- How do you calculate similarity?

Similarity between users: simple way

	Item 1	Item 2	Item 3	Item 4	Item 5
User 1 	8	1	?	2	7
User 2 	2	?	5	7	5

- Only consider items both users have rated
- For each item:
 - Calculate difference in the users' ratings
 - Take the average of this difference over the items

$$\text{Sim}(\text{User1}, \text{User2}) = \frac{\sum_j | \text{rating}(\text{User1}, \text{Item } j) - \text{rating}(\text{User2}, \text{Item } j) |}{\text{Num. of items}}$$

Problem: similarity between users

	Item 1	Item 2	Item 3	Item 4	Item 5
User 1	1	2	3	4	5
User 2	5	4	3	2	1

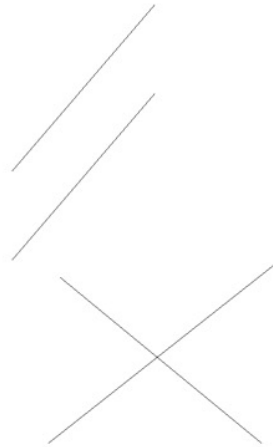
$\text{Sim}(\text{User1}, \text{User2}) = 12/5 = 2.4$

	Item 1	Item 2	Item 3	Item 4	Item 5
User 3	1	2	3	4	5
User 4	4	5	6	7	8

$\text{Sim}(\text{User3}, \text{User4}) = 15/5 = 3$

Better solution

- Use Statistical Correlation Metrics (e.g., Pearson's)
 - These measure how well two data sets fit on a straight line
 - Corrects for grade inflation



Perfect Correlation for User3, User4

Inverse Correlation for User1, User2

Activity 2

- ใน colab

Assignment 11

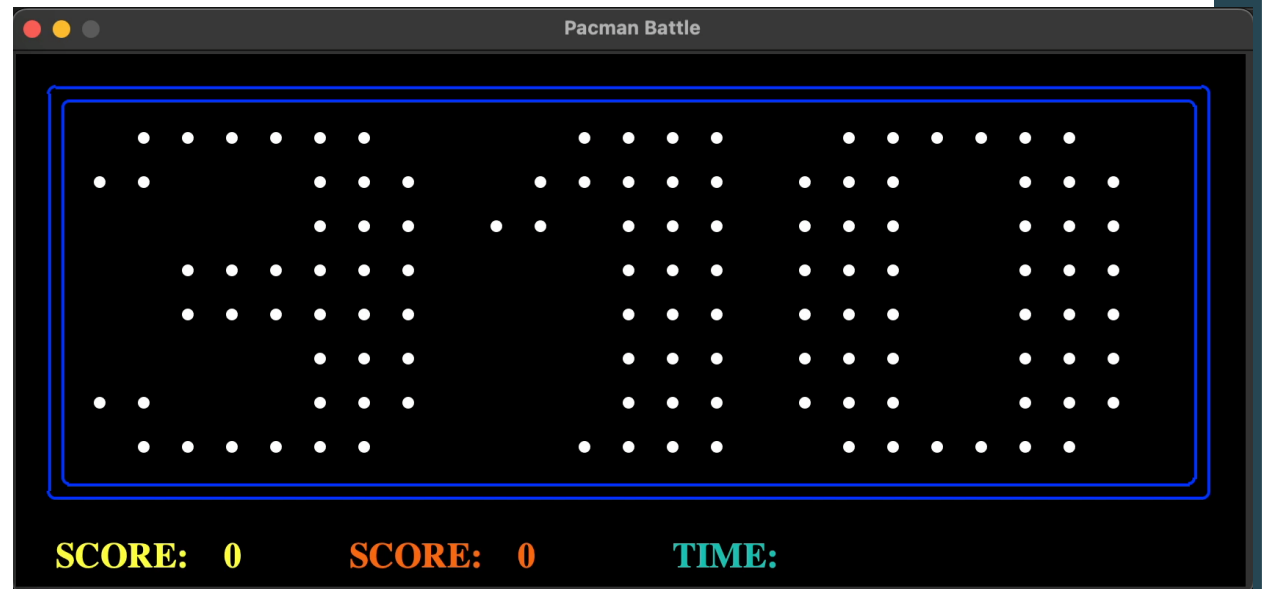
ทำการเพิ่มข้อมูลประเภทของหนัง ในการทำ Recommendation system

- ทำการเพิ่มข้อมูลประเภทของหนังใน column Plot
- ทำการรัน recommendation แบบ content-based filtering ทั้งแบบใช้ TF-IDF และ แบบ Bag of words
- ทดลองหาหนังที่มีเนื้อหาใกล้เคียงกับหนังสองเรื่องนี้ คือ The Godfather และ The Dark Knight

Week 11: Classroom game

- ตอบคำถามใน LEB2 ก่อนสัปดาห์หน้า

AI Project (25%)



Pacman Tournament (คะแนน 25%)

- การแข่งขัน Pacman Tournament วันที่ 16 พ.ค. 2567
- ทีมละ 5 คน อยู่กับเพื่อนๆ ต่าง section ได้
- ทุกทีมจะต้องพัฒนาวิธีการเล่น Pacman ของทีมตัวเอง โดยใช้ code เริ่มต้นจากที่อาจารย์ทำไว้ให้
- เราจะแข่งขันแบบ Tournament คือ ทุกทีมจะแข่งกับทีมอื่น
- ทีมที่ชนะใน Tournament จะได้ 25 คะแนน อันดับที่ 2 จะได้ 22 คะแนน อันดับที่ 3 จะได้ 20 คะแนน หลังจากนั้น คะแนนจะลดไปเรื่อยๆ จนถึง 5 คะแนน
- ไม่ส่ง ไม่ทำ ไม่พัฒนาเพิ่มเติม หรือส่ง code แต่รันไม่ได้ ได้ 0 คะแนน

กติกาการแข่งขัน

- เล่นทีละสองกลุ่ม โดยแต่ละกลุ่มจะได้ Pacman กลุ่มละสี่
- ในเกมจะประกอบด้วยอาหาร และ item ที่ชื่อว่า capsule
 - ถ้า Pacman ของเรากินอาหาร เราจะได้ 10 คะแนน
 - ถ้า Pacman ของเรากิน capsule เราจะมีเวลา 20 วินาที ในการขโมยคะแนนครั้งนี้ จากฝั่งตรงข้าม
- แต่ละเกมมีเวลา 90 วินาทีในการแข่งขัน และแต่ละทีมมีเวลา 20 วินาทีในการคำนวณในแต่ละเกม
- เมื่อหมดเวลา จะถือว่าจบเกม ใครที่ได้คะแนนสูงกว่าจะเป็นผู้ชนะในเกมนั้น
- ถ้าได้คะแนนเท่ากัน ทีมที่ใช้เวลาในการคำนวณน้อยกว่าจะเป็นผู้ชนะ

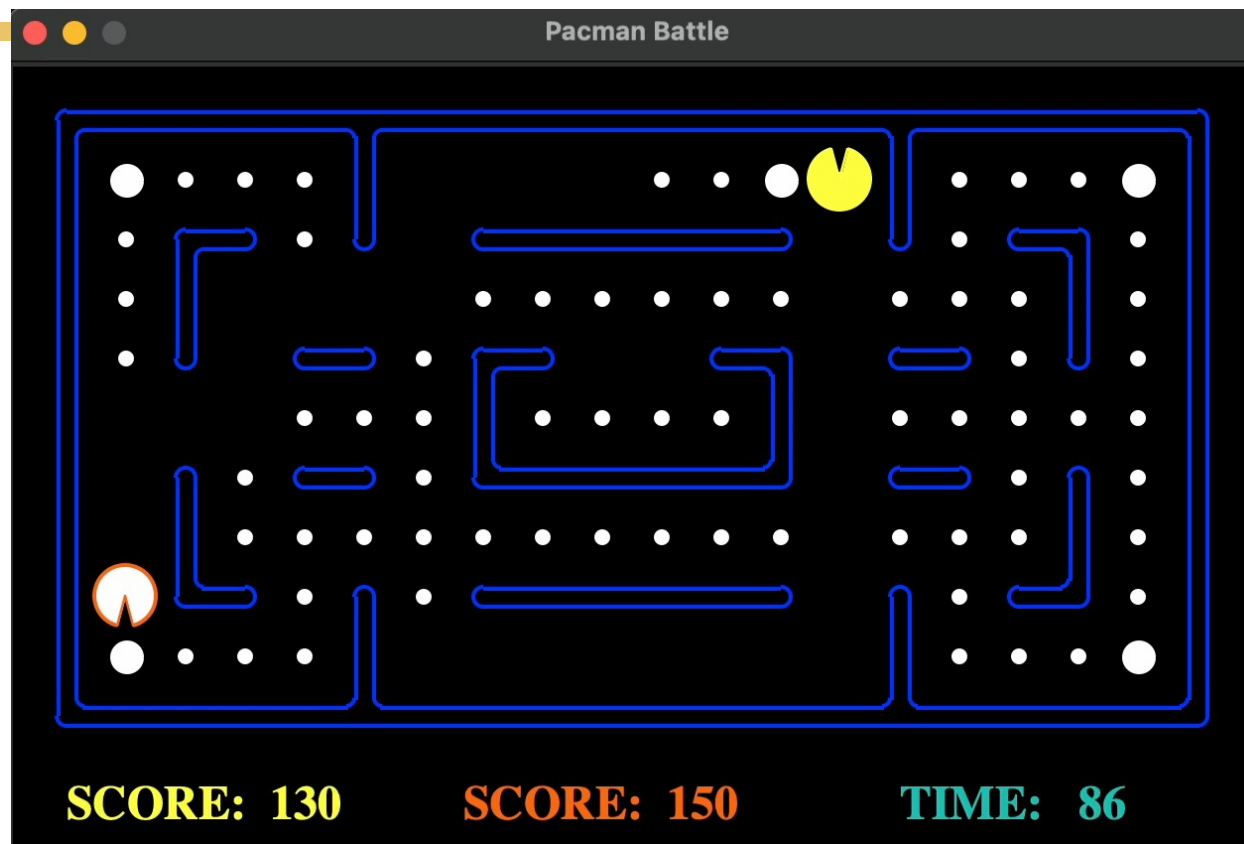
กติกาการแข่งขัน

- ในแต่ละเกม ทีมใดใช้เวลาคำนวณนานเกิน 20 วินาที แพ้ในเกมนั้นทันที
- ใน 1 รอบของการแข่ง จะแข่งทั้งหมด 3 เกม ใน 3 maps ที่แตกต่างกัน
- ทีมที่ชนะ 2 ใน 3 map จะเป็นผู้ชนะ ในรอบนั้น
- ทีมที่ชนะในรอบนั้นจะได้ 1 คะแนน ทีมที่แพ้ได้ 0 คะแนน
- แข่งแบบ tournament คือทุกทีมเจอกันหมด

เกม Pacman

- Python Code ประกอบไปด้วย 9 files ด้วยกัน
- นศ แก่ได้เฉพาะ file ที่ชื่อว่า submission.py
- file อื่นห้ามแก่นะ เดียวจะรัน tournament ไม่ได้
- ส่ง submission.py ภายในวันที่ 14 พ.ค. 2567 ที่ LEB2 ส่งเป็นกลุ่ม
- ส่ง slide นำเสนอว่ามีเทคนิคที่จะเอาชนะคู่ต่อสู้อย่างไร
- ทีมละ 5 คน อยู่กับเพื่อนๆ ต่าง section ได้
- Download all files: <https://github.com/ketnas/pacman-310>

ตัวอย่างการเล่น Pacman



ตัวอย่างการเล่น Pacman

