Week 12-13:
Hybrid Recommender Systems and Evaluation Metrics

CSX4207/ITX4207: Decision Support and Recommender Systems

ITX4287: Selected Topic in Decision Support and Recommender Systems

Asst. Prof. Dr. Rachsuda Setthawong

Asst. Prof. Dr. Rachsuda Setthawong

Objectives

- To understand concept of hybridizing recommender systems
- To understand different types of Hybrid Recommender Systems
- To be able to create simple Hybrid Recommender Systems
- To understand evaluation metrics

Outlines

- Hybrid Recommender Systems
- Types of Hybrid Recommender Systems
  - Monolithic Hybridization Design
  - Parallelized Hybridization Design
  - Pipelined Hybridization Designs
- Evaluation Metrics

Asst. Prof. Dr. Rachsuda Setthawong

Hybrid Recommender Systems

- *hybrida* [lat.]: an object made by combining two different elements

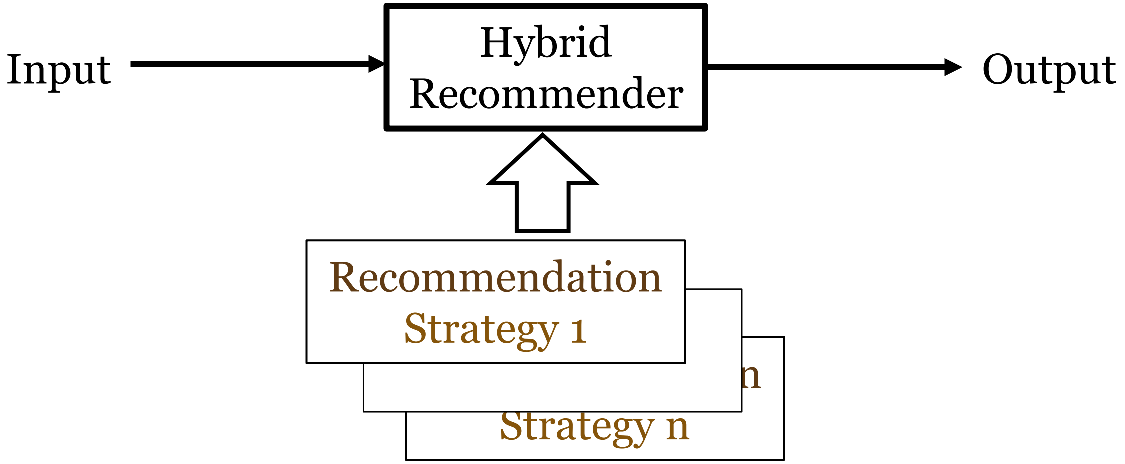
- Main Idea:
  - Cross two (or more) species/implementations.
  - Combine the strengths of the three base techniques to overcome their weak points and problems.
    - Achieve desirable properties not present in parent individuals

Types of Hybrid Recommender Systems

- Monolithic Hybridization Design
  - Feature Combination
  - Feature Augmentation
- Parallelized Hybridization Design
- Pipelined Hybridization Designs

Monolithic Hybridization Design

- Only a single recommendation component

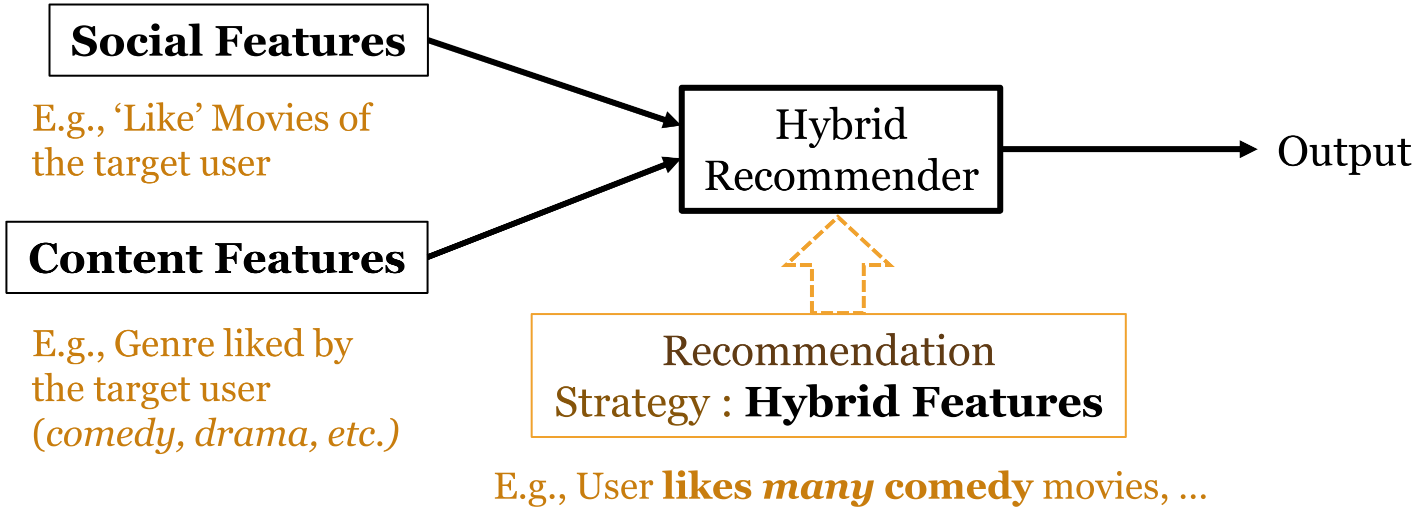


- "Virtual" hybridization
  - Combine ***Features/knowledge sources*** of different paradigms.

Asst. Prof. Dr. Rachsuda Setthawong

Monolithic Hybridization Designs: Feature Combination

- Combine knowledge from several sources, *e.g., ratings, user demographics* and/or explicit requirements in computing similarity.
- "Hybrid" content features:



Asst. Prof. Dr. Rachsuda Setthawong

Example 1 of Feature Combination (1)

- Rule for deriving hybrid features:
  - IF a user bought mainly items of genre X, THEN set the user characteristic ***'User likes many X items'*** to true.
- Input data:

Genre	romance	mystery	mystery	mystery	fiction
User	**Item1**	**Item2**	**Item3**	**Item4**	**Item5**
Alice		1		1	
User1		1	1		1
User2	1	1			1
User3	1		1		
User4					1

Item	Genre
Item1	romance
Item2	mystery
Item3	mystery
Item4	mystery
Item5	fiction

Example 1 of Feature Combination (2)

Genre	*romance*	*mystery*	*mystery*	*mystery*	*fiction*
User	**Item1**	**Item2**	**Item3**	**Item4**	**Item5**
Alice		1		1	
User1		1	1		1
User2	1	1			1
User3	1		1		
User4					1

Rule for deriving hybrid features:

- IF # of purchase > = 2 THEN ‘many’ = true
- IF # of purchase = 1 THEN ‘some’ = true

hybrid features

User	**Alice**	**User1**	**User2**	**User3**	**User4**
User ***likes many mystery*** books	true	true			
User ***likes some mystery*** books			true	true	
User ***likes many romance*** books					
User ***likes some romance*** books			true	true	
User ***likes many fiction*** books					
User ***likes some fiction*** books		true	true		true

Asst. Prof. Dr. Rachsuda Setthawong

Example 1 of Feature Combination (3)

- Who is the most similar user to Alice?

Genre	*romance*	*mystery*	*mystery*	*mystery*	*fiction*
User	**Item1**	**Item2**	**Item3**	**Item4**	**Item5**
Alice		1		1	
User1		1	1		1
User2	1	1			1
User3	1		1		
User4					1

Asst. Prof. Dr. Rachsuda Setthawong

Example 1 of Feature Combination (4)

- Who is the most similar user to Alice *wrt the combined features*?

Genre		*romance*	***mystery***	*mystery*	***mystery***	*fiction*
User	**Item1**	**Item2**	**Item3**	**Item4**	**Item5**	
Alice		1		1		
User1		1	1		1	
User2	1	1			1	
User3	1		1			
User4					1	

Rule for deriving hybrid features:

- IF # of purchase > = 2 THEN 'many' = true
- IF # of purchase = 1 THEN 'some' = true

User	Alice	User1	User2	User3	User4
User ***likes many mystery*** books	true	true			
User ***likes some mystery*** books			true	true	
User ***likes many romance*** books					
User ***likes some romance*** books			true	true	
User ***likes many fiction*** books					
User ***likes some fiction*** books		true	true		true

Asst. Prof. Dr. Rachsuda Setthawong

Example 2 of Feature Combination (1)

- Who is the most similar user to Alice *NOT considering* priority of the features?

User	Navigation Action (f1)	Click on item's detail pages (f2)	Contextual user requirements (f3)	Actual purchase (f4)
Alice	n3, n4	i5	k5	∅
User1	n1, n5	i3, i5	k5	i1
User2	n3, n4	i3, i5, i7	∅	i3
User3	n2, n3, n4	i2, i4, i5	k2, k4	i4

Asst. Prof. Dr. Rachsuda Setthawong

Example 2 of Feature Combination (2)

- Who is the most similar user to Alice *NOT considering* priority of the features?

User	Navigation Action (f1)	Click on item's detail pages (f2)	Contextual user requirements (f3)	Actual purchase (f4)
Alice	n3, n4	i5	k5	∅
User1	n1, n5	i3, i5	k5	i1
User2	n3, n4	i3, i5, i7	∅	i3
User3	n2, n3, n4	i2, i4, i5	k2, k4	i4

Example 2 of Feature Combination (3)

- Suppose that the priority of the features are as follows: (f3, f4) > f2 > f1,
  - who is the most similar user to Alice *considering* the priority of the features?

User	Navigation Action (f1)	Click on item's detail pages (f2)	Contextual user requirements (f3)	Actual purchase (f4)
Alice	n3, n4	i5	k5	∅
User1	n1, n5	i3, i5	k5	i1
User2	n3, n4	i3, i5, i7	∅	i3
User3	n2, n3, n4	i2, i4, i5	k2, k4	i4

Asst. Prof. Dr. Rachsuda Setthawong

Example 2 of Feature Combination (4)

- Suppose that the priority of the features are as follows: (f3, f4) > f2 > f1,
  - who is the most similar user to Alice *considering* the priority of the features?

User	Navigation Action (f1)	Click on item's detail pages (f2)	Contextual user requirements (f3)	Actual purchase (f4)
Alice	n3, n4	i5	k5	∅
User1	n1, n5	i3, i5	k5	i1
User2	n3, n4	i3, i5, i7	∅	i3
User3	n2, n3, n4	i2, i4, i5	k2, k4	i4

Monolithic Hybridization Designs: Feature Augmentation

- Main Idea: apply *more complex transformation steps* to combine several types of inputs.
  - *Augments* the **feature space** of the actual recommender *by* preprocessing its knowledge sources.

Asst. Prof. Dr. Rachsuda Setthawong

An Example of Feature Augmentation:
Content-boosted collaborative filtering (1)

- Predicts a **target user's rating** based **on a collaborative mechanism** *that includes content-based predictions.*
  - What is Alice's rating on item5?

User	User rating on item5 ($v_{User,Item5}$)	Pearson Coeff. ($P_{Alice,User}$)	No. of user ratings (n_{User})	No. of overlapping rating ($n_{Alice,User\_i}$)
Alice	?		40	
User1	4	0.8	14	6
User2	2.2	0.7	55	28

Asst. Prof. Dr. Rachsuda Setthawong

An Example of Feature Augmentation: Content-boosted collaborative filtering (2)

- Step 1: create a pseudo-user-ratings vector $v_{u,i}$:

$$v_{u,i} = \begin{cases} r_{u,i} & : if\ user\ u\ rated\ item\ i \\ c_{u,i} & else\ content-based\ prediction \end{cases}$$

User	User rating on item5 ($v_{User,Item5}$)	Pearson Coeff. ($P_{Alice,User}$)	No. of user ratings (n_{User})	No. of overlapping rating ($n_{Alice,User\_i}$)
Alice	?		40	
User1	4	0.8	14	6
User2	2.2	0.7	55	28

Rating that user1 gives to item5.

Obtained from content-based prediction for User2 on item5.

An Example of Feature Augmentation:

Content-boosted collaborative filtering (3)

- Step 2: compute predictions based on the pseudo ratings:

Self-weighting on
The content-based prediction
on item5

Adjust user's rating based on
Correlation between the target user
and the most similar user(s)

Target user

item

$$rec_{cbcf}(a,i) = \frac{\overbrace{sw_a c_{a,i}} + \overbrace{\sum_{\substack{u=1\\u\neq a}}^{n} hw_{a,u} P_{a,u} v_{u,i}}}{sw_a + \sum_{\substack{u=1\\u\neq a}}^{n} hw_{a,u} P_{a,u}}$$

User's rating on item5
($v_{User,Item5}$)

Pearson Coeff.
($P_{Alice,User}$)

Asst. Prof. Dr. Rachsuda Setthawong

An Example of Feature Augmentation:
Content-boosted collaborative filtering (4)

content-based prediction on item *i*

Assume that $C_{Alice,Item5} = 3$

User	No. of user ratings (n_{User})
Alice	40
User1	14
User2	55

Self-weighting on The content-based prediction on item5

Target user

item

$$rec_{cbcf}(a,i) = \frac{sw_a c_{a,i} + \sum_{u=1, u \neq a}^{n} hw_{a,u} P_{a,u} v_{u,i}}{sw_a + \sum_{u=1, u \neq a}^{n} hw_{a,u} P_{a,u}}$$

Self-weighing factor: (confidence wrt no. of original rating values of the target user *a*)

$$sw_i = \begin{cases} \frac{n_i}{50} \times \max : if \;\; n_i < 50 \\ \max \quad : Otherwise \end{cases}$$

Assume that max = 2 (in the paper)

$$sw_{Alice} = \frac{n_{Alice}}{50} \times max = \frac{40}{50} \times 2 = 1.6$$

Asst. Prof. Dr. Rachsuda Setthawong

An Example of Feature Augmentation: Content-boosted collaborative filtering (5)

Adjust user's rating based on Correlation between the target user and the most similar user(s)

Target user

item

$$rec_{cbcf}(a,i) = \frac{sw_a c_{a,i} + \sum_{u=1, u\neq a}^{n} hw_{a,u} P_{a,u} v_{u,i}}{sw_a + \sum_{u=1, u\neq a}^{n} hw_{a,u} P_{a,u}}$$

User	User rating on item5 ($v_{User,Item5}$)	Pearson Coeff. ($P_{Alice,User}$)
Alice	?	
User1	4	0.8
User2	2.2	0.7

Pseudo-user-ratings (from step 1)

Pearson correlation coeff.

Hybrid correlation weight (calculation shown in the next slide)

Asst. Prof. Dr. Rachsuda Setthawong

Hybrid correlation weight

User	User rating on item5 ($v_{User,Item5}$)	Pearson Coeff. ($P_{Alice,User}$)	No. of user ratings (n_{User})	No. of overlapping rating ($n_{Alice,User}$)
Alice	?		40	
User1	4	0.8	14	6
User2	2.2	0.7	55	28

$$hw_{a,u} = sg_{a,u} + hm_{a,u}$$

Significant Weighting Factor ($sg_{a,u}$)

$$sg_{a,u} = \begin{cases} \frac{n_{a,u}}{50} & : if\ n_{a,u} < 50 \\ 1 & : Otherwise \end{cases}$$

Harmonic Mean Weighting Factor ($hm_{a,u}$)

$$hm_{a,u} = \frac{2m_a m_u}{m_a + m_u}$$

$$m_i = \begin{cases} \frac{n_i}{50} & : if\ n_i < 50 \\ 1 & : Otherwise \end{cases}$$

$$hw_{Alice,User1} = sg_{Alice,User1} + hm_{Alice,User1} = 0.12 + 0.415 = 0.535$$

$$sg_{Alice,User1} = \frac{n_{Alice,User1}}{50} = \frac{6}{50} = 0.12$$

$$hm_{Alice,User1} = \frac{2m_{Alice}m_{User1}}{m_{Alice}+m_{User1}} = \frac{2\times\frac{40}{50}\times\frac{14}{50}}{\frac{40}{50}+\frac{14}{50}} = \frac{2\times0.8\times0.28}{0.8+0.28} = 0.415$$

Asst. Prof. Dr. Rachsuda Setthawong

An Example of Feature Augmentation (6)

- Step 2: compute predictions based on the pseudo ratings:

Target user

item

$$rec_{cbcf}(a,i) = \frac{sw_a c_{a,i} + \sum_{\substack{u=1 \\ u \neq a}}^{n} hw_{a,u} P_{a,u} v_{u,i}}{sw_a + \sum_{\substack{u=1 \\ u \neq a}}^{n} hw_{a,u} P_{a,u}}$$

Self-weighing factor

content-based prediction on item i

Pseudo-user-ratings

Pearson correlation coeff.

Hybrid correlation weight

$c_{Alice,\ Item5}$

$$rec_{cbcf}(Alice, Item5) = \frac{1.6 \times 3 + (0.535 \times 0.8 \times 4 + 1.45 \times 0.7 \times 2.2)}{1.6 + (0.535 \times 0.8 + 1.45 \times 0.7)} = 2.87$$

Asst. Prof. Dr. Rachsuda Setthawong

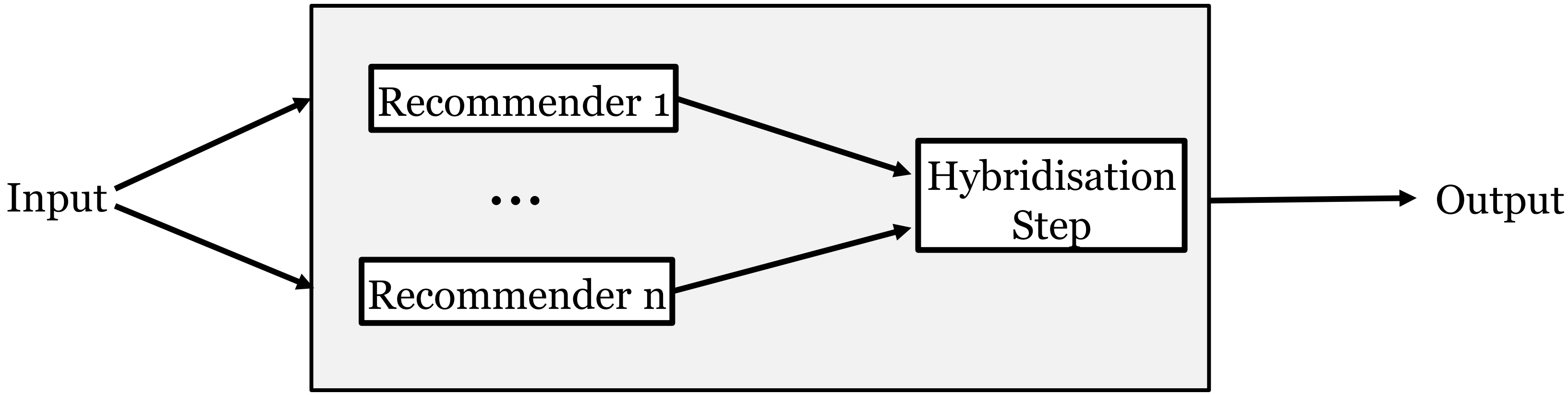
Types of Hybrid Recommender Systems

- Monolithic Hybridization Design
- Parallelized Hybridization Design
  - Weighted
  - Switching
- Pipelined Hybridization Designs

Asst. Prof. Dr. Rachsuda Setthawong

Parallelized Hybridization Design

- **Run** several recommendation algorithms *(in **parallel**)* and **combine** results to generate final recommendations.
- May apply weighting (voting) for result combination.
  - **Approach 1:** feasibly learn weights *dynamically*.
  - **Approach 2:** use ***switching*** if weights of only one algorithm is one; the rest algorithm(s) has zero weight.



Asst. Prof. Dr. Rachsuda Setthawong

Parallelized Hybridization Design: Computing Weighted Recommendations

No. of recommenders

Weight

$$rec_{weighted}(u,i) = \sum_{k=1}^{n} \beta_k \times rec_k(u,i)$$

Weighted recommendation on item *i* for the active user *u*

Recommender 1		
Item1	0.5	1
Item2	0	
Item3	0.3	2
Item4	0.1	3
Item5	0	

Recommender 2		
Item1	0.8	2
Item2	0.9	1
Item3	0.4	3
Item4	0	
Item5	0	

[(0.5*0.5) + (0.5*0.8)]

Recommender weighted (0.5 : 0.5)		
Item1	0.65	1
Item2	0.45	2
Item3	0.35	3
Item4	0.05	4
Item5	0.00	

Note:

1. Item scores must be in the same range.
2. $\Sigma\beta_k=1$

Asst. Prof. Dr. Rachsuda Setthawong

Parallelized Hybridization Design: Weighting Strategies

- Empirical bootstrapping
  - Require historic data.
  - **Generate recommendations** using **different weightings and select the one** with the **best** result**.**
- Dynamic adjustment of weights
  - Initially use *uniform weight distribution.*
  - Iteratively *adjust weights to minimize prediction error.*

Asst. Prof. Dr. Rachsuda Setthawong

Parallelized Hybridization Design: Caution when Applying Weighted (1)

- Suppose that Alice bought items 1 and 4 *(= actual rating = (r_i = 1))*
  - Select weighting with smallest Mean Absolute Error (MAE)

$$MAE = \frac{\sum_{r_i \in R} \sum_{k=1}^{n} \beta_k \times |rec_k(u,i) - r_i|}{|R|}$$

Difference between actual and predicted ratings

Selected weights with best MAE (lowest)

Absolute Errors and MAE

Beta1	Beta2		Predicted rating on Rec1	Predicted rating on Rec2	error	MAE
0.1	0.9	Item1	0.5	0.8	0.23	0.61
		Item4	0.1	0.0	0.99	
0.3	0.7	Item1	0.5	0.8	0.29	0.63
		Item4	0.1	0.0	0.97	
0.5	0.5	Item1	0.5	0.8	0.35	0.65
		Item4	0.1	0.0	0.95	
0.7	0.3	Item1	0.5	0.8	0.41	0.67
		Item4	0.1	0.0	0.93	
0.9	0.1	Item1	0.5	0.8	0.47	0.69
		Item4	0.1	0.0	0.91	

low

high

Asst. Prof. Dr. Rachsuda Setthawong

Parallelized Hybridization Design: Caution when Applying Weighted (2)

- Suppose that Alice bought items 1 and 4 (= *actual rating* = (r_i = 1))
  - Select weighting with smallest Mean Absolute Error (MAE)

Absolute difference between actual and predicted ratings

$$MAE = \frac{\sum_{r_i \in R} \sum_{k=1}^{n} \beta_k \times |rec_k(u,i) - r_i|}{|R|}$$

Error_item1[beta(0.1, 0.9)] =(0.1 x |0.5-1|) + (0.9 x |0.8-1|) = 0.23

Absolute Errors and MAE

Beta1	Beta2		Predicted rating on Rec1	Predicted rating on Rec2	error	MAE
0.1	0.9	Item1	0.5	0.8	0.23	0.61
		Item4	0.1	0.0	0.99	

Asst. Prof. Dr. Rachsuda Setthawong

Parallelized Hybridization Design: Caution when Applying Weighted (3)

- Suppose that Alice bought items 1 and 4 *(= actual rating = (ri = 1))*
  - Select weighting with smallest Mean Absolute Error (MAE)

Absolute difference between actual and predicted ratings

$$MAE = \frac{\sum_{r_i \in R} \sum_{k=1}^{n} \beta_k \times |rec_k(u,i) - r_i|}{|R|}$$

MAE(beta(0.1,0.9)) = error_item1 + error_item4 =
{ [(0.1 x |0.5-1|) + (0.9 x |0.8-1|)] + [(0.1 x |0.1-1|) + (0.9 x |0.0-1|)] } / 2
= (0.23 + 0.99) / 2 = 0.61

Absolute Errors and MAE

Beta1	Beta2		Predicted rating on Rec1	Predicted rating on Rec2	error	MAE
0.1	0.9	Item1	0.5	0.8	0.23	0.61
		Item4	0.1	0.0	0.99	

Parallelized Hybridization Design: Switching

- $\exists_1 k : 1 \ldots n \;\; rec_{switching}(u, i) = rec_k(u, i)$
- Equivalent to dynamic weights when one β equals 1 and the rest βs equal 0.
- An example:

IF *too few ratings in the system* THEN

use knowledge-based

ELSE

use collaborative-based

Asst. Prof. Dr. Rachsuda Setthawong

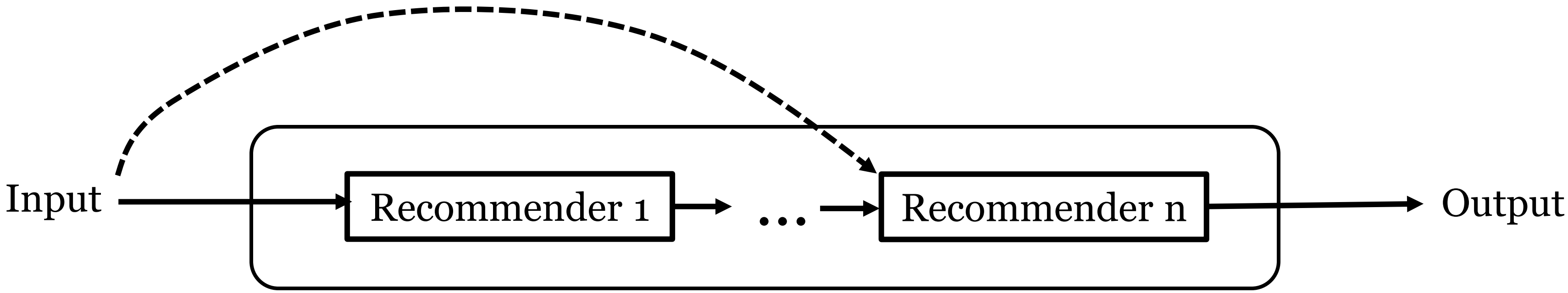
Types of Hybrid Recommender Systems

- Monolithic Hybridization Design
- Parallelized Hybridization Design
- Pipelined Hybridization Designs
  - Cascade

Asst. Prof. Dr. Rachsuda Setthawong

Pipelined Hybridization Designs

- Sequentially run recommender systems such that a successor alters results from its predecessor.
- Two designs
  - Cascade
  - Meta-level



Asst. Prof. Dr. Rachsuda Setthawong

Pipelined Hybridization Designs: Cascade (1)

- Generating successor's recommendations:

$$rec_{cascade}(u,i) = rec_n(u,i)$$

where $\forall k \geq 2$

$$rec_k(u,i) = \begin{cases} rec_k(u,i) & : rec_{k-1}(u,i) \neq 0 \\ 0 & : \text{else} \end{cases}$$

- ***Subsequent*** recommender ***either alters*** score of its predecessor's items ***or discards*** them.
  - An item will be ***suggested*** by the k^{th} technique ***only if*** the $(k\text{-}1)^{th}$ technique ***also*** assigned a ***nonzero score*** to it.

Asst. Prof. Dr. Rachsuda Setthawong

Pipelined Hybridization Designs: Cascade (2)

Recommender 1		
Item1	0.5	1
Item2	0	
Item3	0.3	2
Item4	0.1	3
Item5	0	

Recommender 2		
Item1	0.8	2
Item2	0.9	1
Item3	0.4	3
Item4	0	
Item5	0	

Recommender cascaded (rec1, rec2)		
Item1	0.5	
Item2	0	
Item3	0.3	
Item4	0.1	
Item5	0	

- Examples of applying cascade RS:
  - Recommender 1 discards some items.
  - Recommender 2 re-calculating their score.

Pipelined Hybridization Designs: Cascade (3)

Recommender 1		
Item1	0.5	1
Item2	0	
Item3	0.3	2
Item4	0.1	3
Item5	0	

Recommender 2		
Item1	0.8	2
Item2	0.9	1
Item3	0.4	3
Item4	0	
Item5	0	

Recommender cascaded (rec1, rec2)		
Item1	0.8	
Item2	0	
Item3	0.3	
Item4	0.1	
Item5	0	

- Examples of applying cascade RS:
  - Recommender 1 discards some items.
  - Recommender 2 re-calculating their score.

Asst. Prof. Dr. Rachsuda Setthawong

Pipelined Hybridization Designs: Cascade (4)

Recommender 1		
Item1	0.5	1
Item2	0	
Item3	0.3	2
Item4	0.1	3
Item5	0	

Recommender 2		
Item1	0.8	2
Item2	0.9	1
Item3	0.4	3
Item4	0	
Item5	0	

Recommender cascaded (rec1, rec2)		
Item1	0.8	
Item2	0	No change
Item3	0.3	
Item4	0.1	
Item5	0	

- Examples of applying cascade RS:
  - Recommender 1 discards some items.
  - Recommender 2 re-calculating their score.

Asst. Prof. Dr. Rachsuda Setthawong

Pipelined Hybridization Designs: Cascade (5)

Recommender 1		
Item1	0.5	1
Item2	0	
Item3	0.3	2
Item4	0.1	3
Item5	0	

Recommender 2		
Item1	0.8	2
Item2	0.9	1
Item3	0.4	3
Item4	0	
Item5	0	

Recommender cascaded (rec1, rec2)		
Item1	0.8	
Item2	0	
Item3	0.4	
Item4	0.1	
Item5	0	

- Examples of applying cascade RS:
  - Recommender 1 discards some items.
  - Recommender 2 re-calculating their score.

Asst. Prof. Dr. Rachsuda Setthawong

Pipelined Hybridization Designs: Cascade (6)

Recommender 1		
Item1	0.5	1
Item2	0	
Item3	0.3	2
Item4	0.1	3
Item5	0	

Recommender 2		
Item1	0.8	2
Item2	0.9	1
Item3	0.4	3
Item4	0	
Item5	0	

Recommender cascaded (rec1, rec2)		
Item1	0.8	
Item2	0	
Item3	0.4	
Item4	0.0	
Item5	0	

- Examples of applying cascade RS:
  - Recommender 1 discards some items.
  - Recommender 2 re-calculating their score.

Pipelined Hybridization Designs: Cascade (7)

Recommender 1		
Item1	0.5	1
Item2	0	
Item3	0.3	2
Item4	0.1	3
Item5	0	

Recommender 2		
Item1	0.8	2
Item2	0.9	1
Item3	0.4	3
Item4	0	
Item5	0	

Recommender cascaded (rec1, rec2)		
Item1	0.8	
Item2	0	
Item3	0.4	
Item4	0.0	
Item5	0	No change

- Examples of applying cascade RS:
  - Recommender 1 discards some items.
  - Recommender 2 re-calculating their score.

Asst. Prof. Dr. Rachsuda Setthawong

Pipelined Hybridization Designs: Cascade (8)

Recommender 1		
Item1	0.5	1
Item2	0	
Item3	0.3	2
Item4	0.1	3
Item5	0	

Recommender 2		
Item1	0.8	2
Item2	0.9	1
Item3	0.4	3
Item4	0	
Item5	0	

Recommender cascaded (rec1, rec2)		
Item1	0.8	**1**
Item2	0	
Item3	0.4	**2**
Item4	0.0	
Item5	0	

New ranks

- Examples of applying cascade RS:
  - Recommender 1 discards some items.
  - Recommender 2 re-calculating their score.

Outlines

- Hybrid Recommender Systems
- Types of Hybrid Recommender Systems
  - Monolithic Hybridization Design
  - Parallelized Hybridization Design
  - Pipelined Hybridization Designs
- Evaluation Metrics

Asst. Prof. Dr. Rachsuda Setthawong

Methodology

- N-fold cross-validation:
  - A stratified random selection technique
    - **(N-1) out of N folds** of user profiles is used ***for model building***.
    - **1 out of N folds** of user profiles is used ***for evaluation***.
    - **Repeat** them for **N times** and determine the ***average results***

Row	UserID	MovieID	Rating
1	234	110	5
2	234	151	5
3	234	260	3
4	234	376	5
5	234	539	4[a]
6	234	590	5
7	234	649	1
8	234	719	5[a]
9	234	734	3
10	234	736	2

Note: ‘a’ = randomly selected ratings for testing.

Asst. Prof. Dr. Rachsuda Setthawong

A Demonstration of 5-fold Cross-validation

	Training Set				Test Set	Measure e.g., RMSE
Round 1	r1, r2	r4, r7	r3, r8	r6, r10	r5, r9	$RMSE_{round1} = 0.5$
Round 2	r4, r7	r3, r8	r6, r10	r5, r9	r1, r2	$RMSE_{round2} = 0.2$
Round 3	r3, r8	r6, r10	r5, r9	r1, r2	r4, r7	$RMSE_{round3} = 0.1$
Round 4	r6, r10	r5, r9	r1, r2	r4, r7	r3, r8	$RMSE_{round4} = 0.2$
Round 5	r5, r9	r1, r2	r4, r7	r3, r8	r6, r10	$RMSE_{round5} = 0.1$

Average RMSE = 0.22

Evaluation Tasks

- Prediction Task
  - Compute a ***missing rating*** in the user/item matrix.
  - Require Likert scale ratings.
- Classification Task
  - Select ***a ranked list of n items relevant*** for the user.
  - *Transform* Likert scale ratings *into* ***relevant/irrelevant items***.

Metrics

- Accuracy of predictions
- Accuracy of classifications
- Accuracy of ranks

Accuracy of Predictions

- Mean absolute error

$$MAE = \frac{\sum_{u \in U} \sum_{i \in testset_u} |rec(u,i) - r_{u,i}|}{\sum_{u \in U} |testset_u|}$$

- Normalized MAE

$$NMAE = \frac{MAE}{r_{max} - r_{min}}$$

Asst. Prof. Dr. Rachsuda Setthawong

Accuracy of Classifications (1)

- Precision (= TP / (TP + FP))

$$P_u = \frac{|hits_u|}{|recset_u|}$$

$|hitS_u|$ = the number of correctly recommended relevant items for user u

- Recall (= TP / (TP + FN))

$$R_u = \frac{|hits_u|}{|testset_u|}$$

Additional resource: https://en.wikipedia.org/wiki/Precision_and_recall

Asst. Prof. Dr. Rachsuda Setthawong

Example

- Top-20 suggested items for user 234 *((Item,Rating) tuples)*:

Top 20

Recset_{234} = { (912, 4.8), (47, 4.5), (263, 4.4), **(539, 4.1)**, (348, 4), …, **(719, 3.8)** }

Top 3: (912, 4.8), (47, 4.5), (263, 4.4)

Top 5: (912, 4.8), (47, 4.5), (263, 4.4), (539, 4.1), (348, 4)

A Test Set

Row	UserID	MovieID	Rating
1	234	**539**	4
2	234	**719**	5

- **MAE** = [|4.1 – 4| + |3.8 – 5|] / 2 = (0.1 + 1.2) / 2 = 0.65

- **Top 3** ranked items:
  - $P_{234} = 0/3 = 0$
  - $R_{234} = 0/2 = 0$

- **Top 5** ranked items:
  - $P_{234} = 1/5 = 0.2$
  - $R_{234} = 1/2 = 0.5$

- P_{234} and R_{234} for **Top 20** ranked items?

Accuracy of Classifications (2)

- F1 (F-measure)

$$F1 = \frac{2 \times P \times R}{P + R}$$

- Hit rate (focus on # of users who get at least 1 matched item)

$$hitrate_u = \begin{cases} 1 & if\ hit_u > 0 \\ 0 & otherwise \end{cases}$$

where,

hit_u = # of correctly recommended <u>relevant</u> items for user u

Accuracy of Ranks (*rankscore$_u$'*)

$$rankscore'_u = \frac{rankscore_u}{rankscore_u^{max}}$$

$$rankscore_u = \sum_{i \in hit_u} \frac{1}{2^{\frac{rank(i)-1}{\alpha}}}$$

$$rankscore_u^{max} = \sum_{i \in testset_u} \frac{1}{2^{\frac{idx(i)-1}{\alpha}}}$$

where,

rank(i) = the *position of item i* in the user's recommendation list

α = the *half-life* of utilities (decrease by half for each rank)

idx(i) = assigning a value $\in \{1, \ldots, |testset_u|\}$ corresponding to the order of rank(i)

Asst. Prof. Dr. Rachsuda Setthawong

Example

20 (*ItemID, Rating*) tuples suggested for user 234:

$Recset_{234}$ = { (912, 4.8), (47, 4.5), (263, 4.4), **(539, 4.1)**, (348, 4), …, **(719, 3.8)** }

Rank #4

Rank #20

- Assume, $\alpha = 10$

$$rankscore'_u = \frac{rankscore_u}{rankscore_u^{max}}$$

$$rankscore_u = \sum_{i \in hit_u} \frac{1}{2^{\frac{rank(i)-1}{\alpha}}}$$

$$rankscore_{234} = \frac{1}{2^{\frac{4-1}{10}}} + \frac{1}{2^{\frac{20-1}{10}}} = 1.08$$

$$rankscore'_{234} = \frac{1.08}{1.93} = 0.56$$

$$rankscore_u^{max} = \sum_{i \in testset_u} \frac{1}{2^{\frac{idx(i)-1}{\alpha}}}$$

$$rankscore_{234}^{max} = \frac{1}{2^{\frac{1-1}{10}}} + \frac{1}{2^{\frac{2-1}{10}}} = 1.93$$

Accuracy of Ranks (*$liftindex_u$*)

- Assume that the **ranked list** is *divided into ten equal deciles*.
- *Count the number of hits in each decile* as $S_{1,u}, S_{2,u}, \ldots, S_{10,u}$, where

$$\sum_{i=1}^{10} S_i = hits_u$$

$$liftindex_u = \begin{cases} \dfrac{1 \cdot S_{1,u} + 0.9 \cdot S_{2,u} + \cdots + 0.1 \cdot S_{10,u}}{\sum_{i=1}^{10} S_{i,u}} & if\ hit_u > 0 \\ 0 & otherwise \end{cases}$$

where,

hit_u = # of correctly recommended <u>relevant</u> items for user u

Example

20 Item/Rating tuples suggested for user 234:

$Recset_{234}$ = { (912, 4.8), (47, 4.5), (263, 4.4), **(539, 4.1)**, (348, 4), …, **(719, 3.8)** }

$S_{1,u}$: (912, 4.8), (47, 4.5)

$S_{2,u}$: (263, 4.4), **(539, 4.1)**

…

$S_{10,u}$: …, **(719, 3.8)**

$$liftindex_u = \begin{cases} \dfrac{1 \cdot S_{1,u} + 0.9 \cdot S_{2,u} + \cdots + 0.1 \cdot S_{10,u}}{\sum_{i=1}^{10} S_{i,u}} & if\ hit_u > 0 \\ 0 & otherwise \end{cases}$$

$$liftindex_{234} = \frac{0.9 \cdot \underbrace{\boxed{1}}_{S_{2,u}} + 0.1 \cdot \underbrace{\boxed{1}}_{S_{10,u}}}{2} = 0.5$$

Asst. Prof. Dr. Rachsuda Setthawong

Discounted Cumulative Gain (DCG)

- **Assumption:** The ***highly relevant*** documents are ***more useful*** than moderately relevant documents.

- DCG measures ***ranking quality*** that assesses the recommended list provided by a recommendation engine

$$DCG = \sum_{i=1}^{n} \frac{relevance_i}{log_2(i+1)}$$

Document relevant score of the ordered recommended item$_i$

- E.g., score_recset$_A$ = [2,3,2,3,1] and score_recset$_B$ = [3,3,2,1,2]

$$DCG_A = \frac{2}{log_2(1+1)} + \frac{3}{log_2(2+1)} + \frac{2}{log_2(3+1)} + \frac{3}{log_2(4+1)} + \frac{1}{log_2(5+1)} = 6.57$$

$$DCG_B = \frac{3}{log_2(1+1)} + \frac{3}{log_2(2+1)} + \frac{2}{log_2(3+1)} + \frac{1}{log_2(4+1)} + \frac{2}{log_2(5+1)} = 7.09$$

Ref: https://towardsdatascience.com/evaluate-your-recommendation-engine-using-ndcg-759a851452d1
https://medium.com/swlh/rank-aware-recsys-evaluation-metrics-5191bba16832

Asst. Prof. Dr. Rachsuda Setthawong

Normalized Discounted Cumulative Gain (NDCG)

$$NDCG = \frac{DCG}{iDCG}$$

- where, iDCG = DCG of the ideal order
- NDCG $\in$ [0, 1]
- E.g., score_recset$_A$ = [2,3,2,3,1] and ideal_score_recset$_A$ = [3,3,2,2,1]

$$DCG_A = \frac{2}{log_2(1+1)} + \frac{3}{log_2(2+1)} + \frac{2}{log_2(3+1)} + \frac{3}{log_2(4+1)} + \frac{1}{log_2(5+1)} = 6.57$$

$$iDCG_A = \frac{3}{log_2(1+1)} + \frac{3}{log_2(2+1)} + \frac{2}{log_2(3+1)} + \frac{2}{log_2(4+1)} + \frac{1}{log_2(5+1)} = 7.14$$

$$NDCG = \frac{6.57}{7.14} = 0.92$$

Note: use **Mean NDCG** of all test users' recommended list to evaluate the performance of the recommendation algorithm

Ref: https://towardsdatascience.com/evaluate-your-recommendation-engine-using-ndcg-759a851452d1
https://medium.com/swlh/rank-aware-recsys-evaluation-metrics-5191bba16832