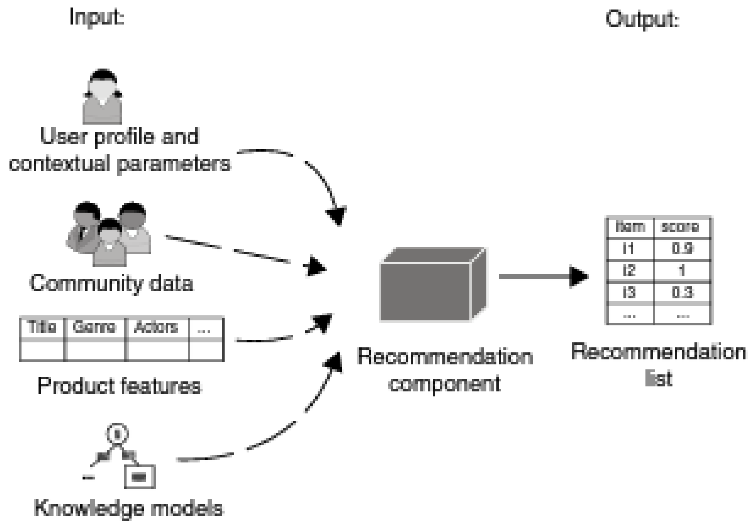
Hybrid Recommendation

Topic 7

Hybrid Recommendation

- Because the three recommendation approaches rely on different sources of information and follow different paradigms with its pros and cons, we can build hybrid systems to combine strengths of different approaches and overcome some of the shortcomings.



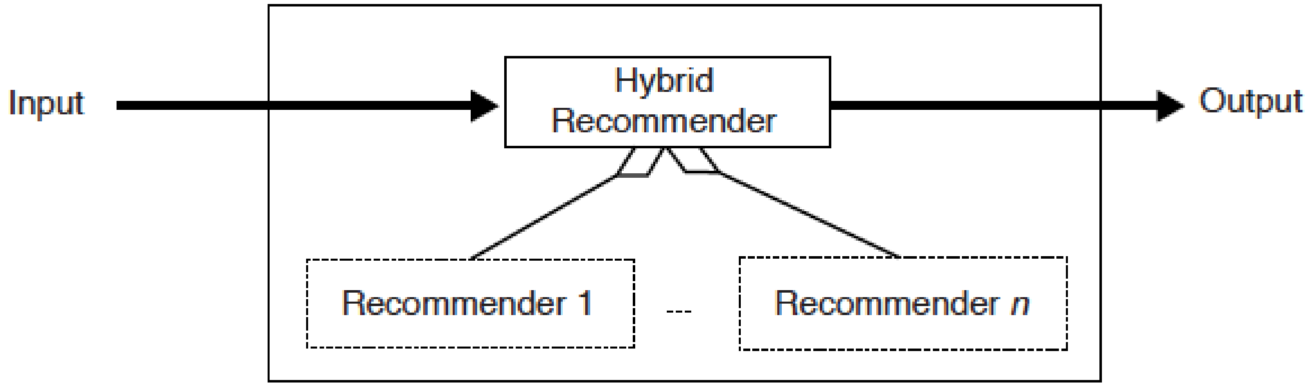
Paradigm	User profile and contextual parameters	Community data	Product features	Knowledge models
Collaborative	Yes	Yes	No	No
Content-based	Yes	No	Yes	No
Knowledge-based	Yes	No	Yes	Yes

Hybridization Designs

- Monolithic hybridization
- Parallelized hybridization
- Pipelined hybridization

Monolithic Hybridization Design

- Incorporates aspects of several recommendation strategies in one algorithm implementation.
- It consists of a single recommender component that integrates multiple approaches by preprocessing and combining several knowledge sources.



Monolithic Hybridization Design

- Feature Combination Hybrids
  - Uses a diverse range of input data and combine them
  - Example

Table 5.2. *Community and product knowledge.*

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

Monolithic Hybridization Design

- Feature Combination Hybrids

Table 5.4. *Different types of user feedback.*

User	R_{nav}	R_{view}	R_{ctx}	R_{buy}
Alice	n_3, n_4	i_5	k_5	$\emptyset$
User1	n_1, n_5	i_3, i_5	k_5	i_1
User2	n_3, n_4	i_3, i_5, i_7	$\emptyset$	i_3
User3	n_2, n_3, n_4	i_2, i_4, i_5	k_2, k_4	i_4

Monolithic Hybridization Design

- Feature Augmentation Hybrids
  - In contrast with feature combination, this hybrid does not simply combine and preprocess several types of input, but rather applies more complex transformation steps.
  - The output of a contributing recommender system augments the feature space of the actual recommender system by preprocessing its knowledge sources.
  - Example
    - Melville et al. (2002)

Monolithic Hybridization Design

User	$v_{User,Item5}$	$P_{Alice,User}$	n_{User}	$n_{Alice,User}$
Alice	?		40	
User1	4	0.8	14	6
User2	2.2	0.7	55	28

$$hw_{a,u} = sg_{a,u} + hm_{a,u}, \text{ where}$$

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

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

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

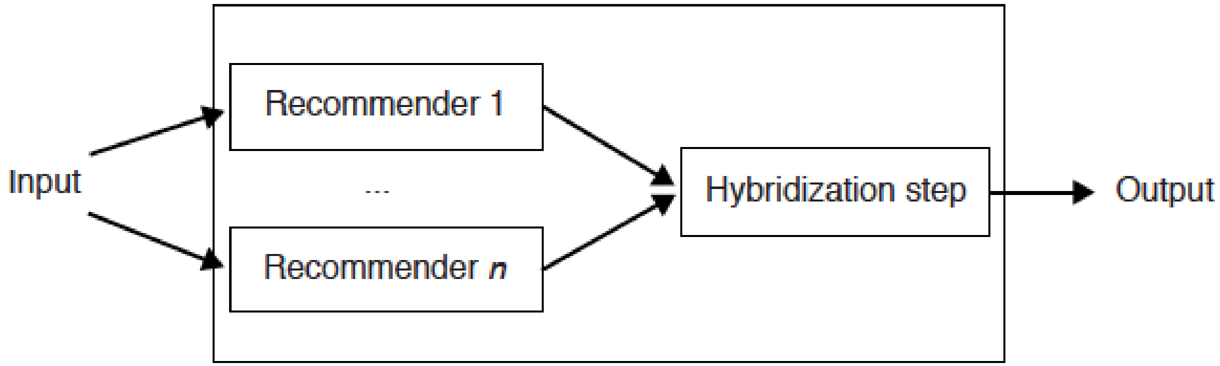
$$sw_i = \begin{cases} \frac{n_i}{50} \times max & : \text{ if } n_i < 50 \\ max & : \text{ else} \end{cases}$$

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

$$\frac{1.6 \times 3 + (0.535 \times 0.8 \times 4 + 1.45 \times 0.7 \times 2.2)}{1.6 + (0.535 \times 0.8 + 1.45 \times 0.7)} = \frac{8.745}{3.043} = 2.87$$

Parallelized Hybridization Design

- Employs several recommenders side by side and employs a specific hybridization mechanism to aggregate their outputs.
- It operates independently of one another and produce separate recommendation lists. Then, in a subsequent hybridization step, their output is combined into a final set of recommendations.



Parallelized Hybridization Design

- Mixed Hybrids

  - Combines the results of different recommender systems at the level of the user interface, in which results from different techniques are presented together.

- Weighted Hybrids

  - Combines the recommendations of two or more recommendation systems by computing weighted sums of their scores.

- Switching Hybrids

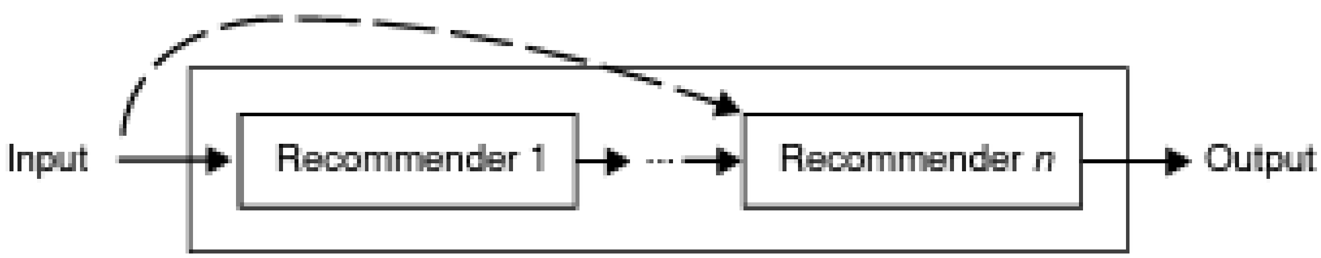
  - Requires an oracle that decides which recommender should be used in a specific situation, depending on the user profile and/or the quality of recommendation results.

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

item	rec_1 score	rec_1 rank	rec_2 score	rec_2 rank	rec_w score	rec_w rank
Item1	0.5	1	0.8	2	0.65	1
Item2	0		0.9	1	0.45	2
Item3	0.3	2	0.4	3	0.35	3
Item4	0.1	3	0		0.05	
Item5	0		0		0	

Pipelined Hybridization Design

- Implements a staged process in which several techniques sequentially build on each other before the final one produces recommendations for the user.
- The output of one recommender becomes part of the input of the subsequent one. Optionally, the subsequent recommender components may use parts of the original input data too.
- A preceding component may either preprocess input data to build a model that is exploited by the subsequent stage or deliver a recommendation list for further refinement.



Pipelined Hybridization Design

- Cascade Hybrids
  - Based on a sequence order of techniques, in which each succeeding recommender only refines the recommendations of its predecessor.
  - The recommendation list of the successor technique is thus restricted to items that were also recommended by the preceding technique.
- Meta-level Hybrids
  - In a meta-level hybridization design, one recommender builds a model that is exploited by the principal recommender to make recommendations.