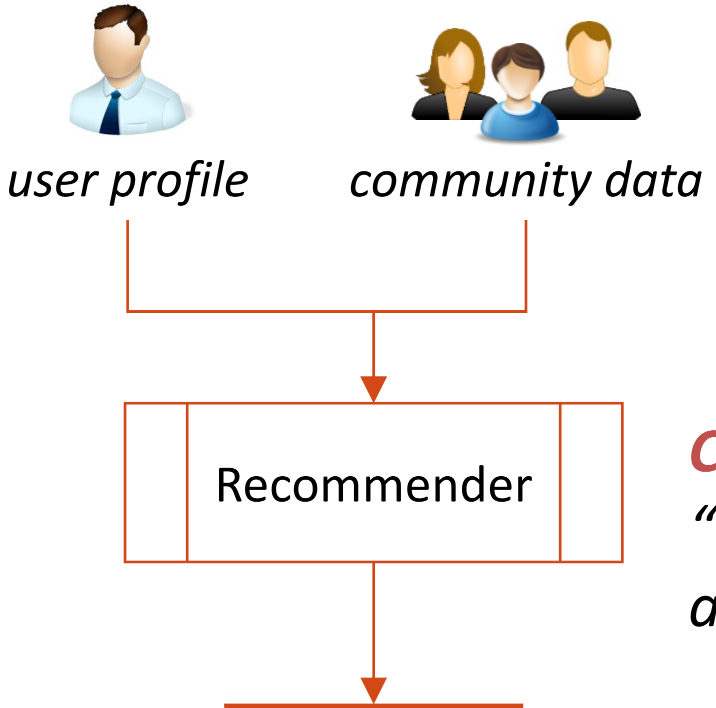
UFMG UNIVERSIDADE FEDERAL DE MINAS GERAIS

Recommender Systems

Hybridization

Rodrygo L. T. Santos
rodrygo@dcc.ufmg.br

How to recommend?



item	score
1	0.7
2	0.3
…	…

Collaborative filtering

“tell me what’s popular among my peers”

How to recommend?

user profile

title	genre	year

item features

Recommender

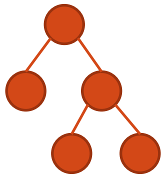
item	score
1	0.7
2	0.3
…	…

Content-based

"show me more of the same what I've liked"

How to recommend?

user profile



knowledge models

Recommender

Knowledge-based
"tell me what fits based on my needs"

item	score
1	0.7
2	0.3
…	…

How to recommend?

Collaborative

- *Good accuracy and discovery*
- *Poor with sparse ratings*

Content-based

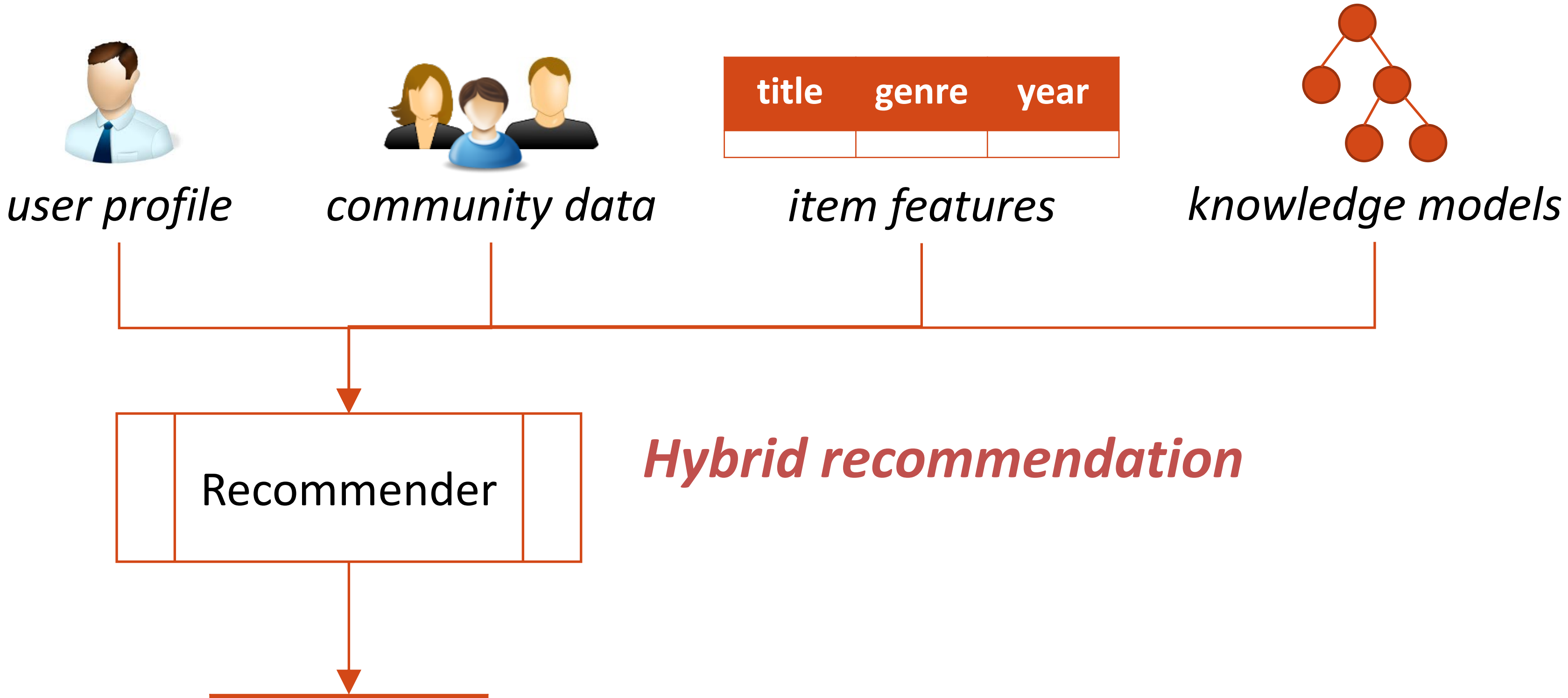
- *No need for item ratings*
- *Poor discovery*

Knowledge-based

- *No need for user or item ratings*
- *Costly knowledge acquisition*

NO
SILVER
BULLET

How to recommend?



Hybrid recommendation

item	score
1	0.7
2	0.3
…	…

Hybrid recommendation

Hybrid: *“formed by combining two or more things”*

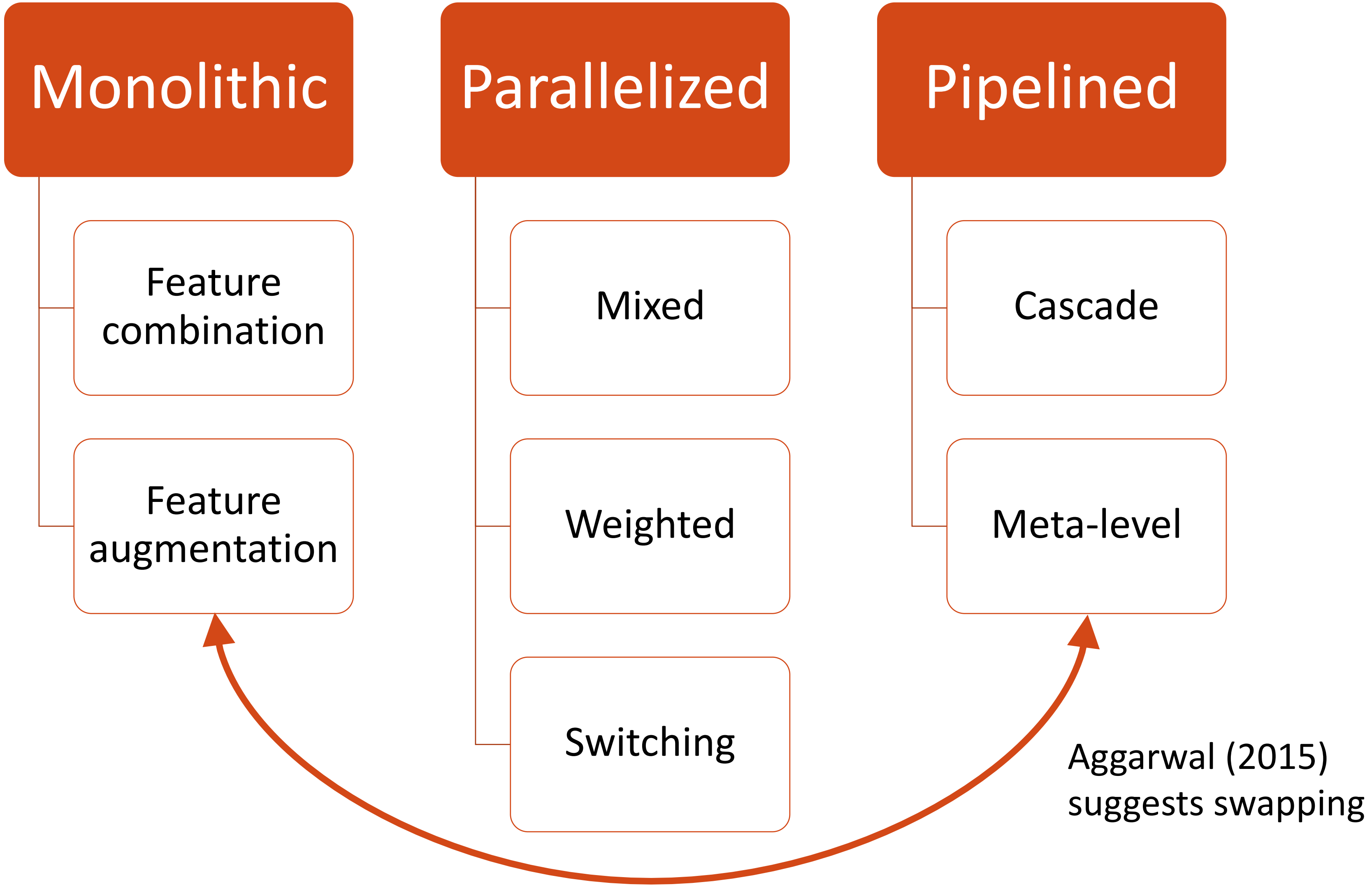
- Incorporate desirable properties
- Avoid some of the shortcomings

No standard hybridization approach

- Actually, there is a large variety!

No established winner

Hybridization designs [Burke, UMUAI 2002]



Hybridization designs [Agarwal, 2015]

Monolithic
- Feature combination
- Meta-level

Parallelized
- Mixed
- Weighted
- Switching

Pipelined
- Cascade
- Feature augmentation

Ensembles

"custom-built" recommenders

"off-the-shelf" recommenders

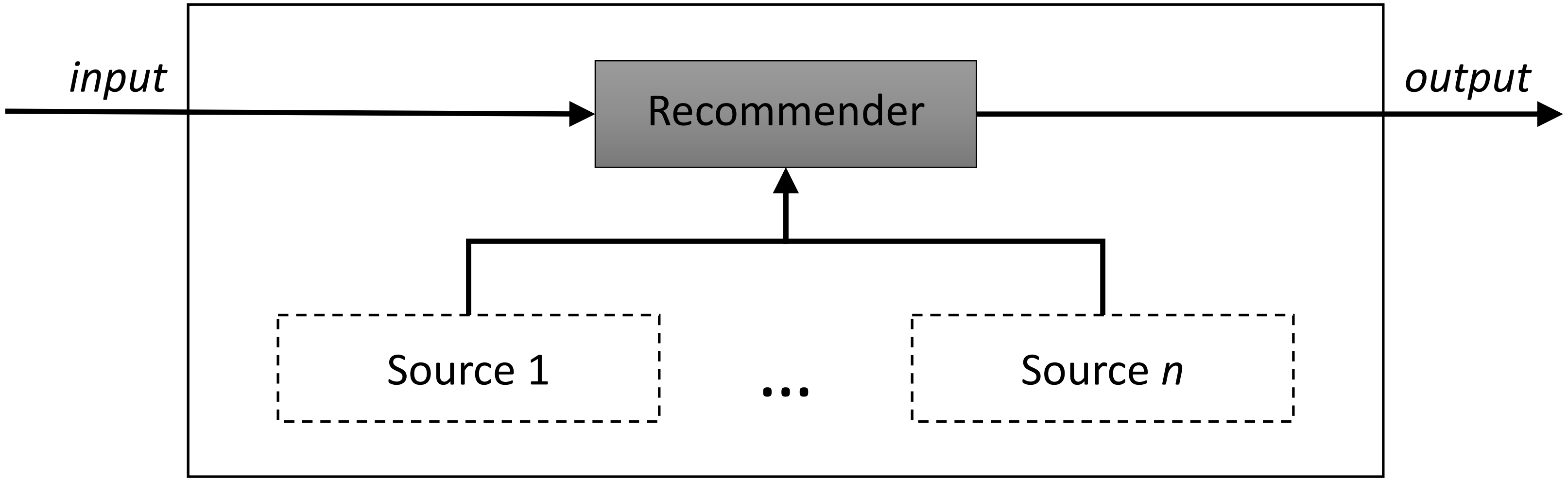
Hybridization designs [Burke, UMUAI 2002]

- Monolithic
  - Feature combination
  - Feature augmentation
- Parallelized
  - Mixed
  - Weighted
  - Switching
- Pipelined
  - Cascade
  - Meta-level

Monolithic hybridization

One recommender

- Multiple data sources



Feature combination

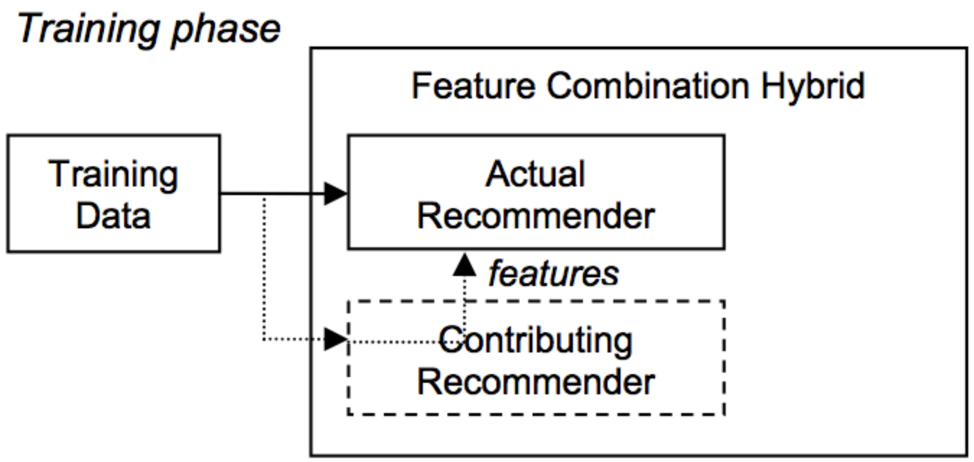
Monolithic

Feature combination

Feature augmentation

Combine data sources

- Recommender uses ***raw*** features of another type



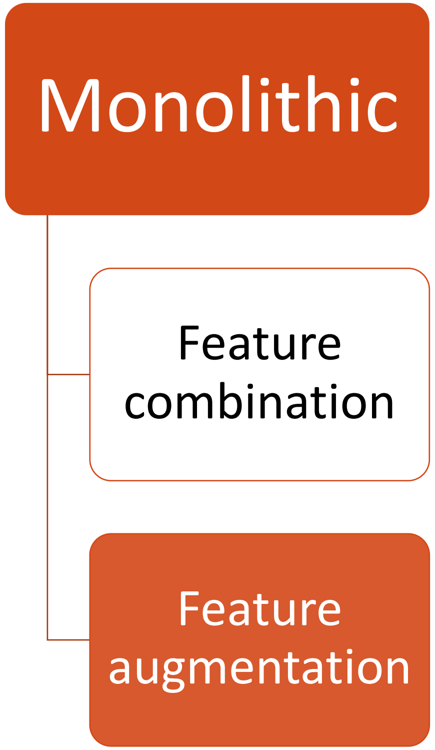
Feature combination

- Monolithic
  - **Feature combination**
  - Feature augmentation

CF/CB movie recommender [Basu et al., AAAI 1998]

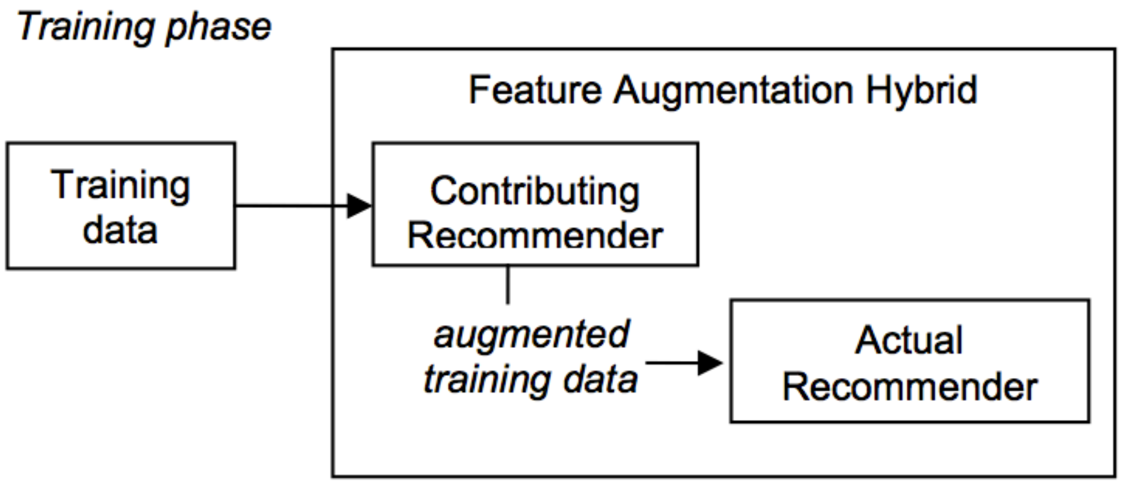
- CB as a contributing recommender
  - Represent users based on movie genres
- CF as the actual recommender
  - Identify neighbor users
  - Aggregate neighbors' preferences

Feature augmentation



Augment data sources

- Recommender uses ***generated*** features



Feature augmentation

- Monolithic
  - Feature combination
  - Feature augmentation

CB/CF movie recommender [Melville et al., AAAI 2002]

- CB as a contributing recommender
  - Augment rating matrix with content-based predictions
- CF as the actual recommender
  - Identify neighbor users (based on actual ratings)
  - Aggregate neighbors' preferences, giving more emphasis on actual ratings and less emphasis on pseudo-ratings

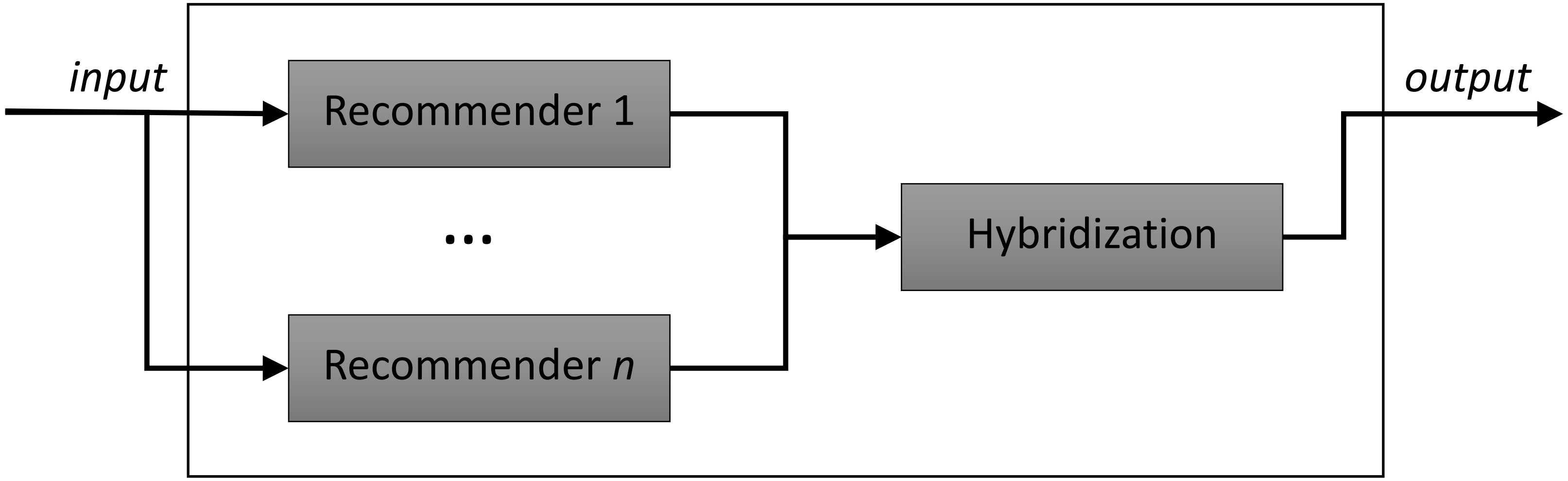
Hybridization designs [Burke, UMUAI 2002]

- Monolithic
  - Feature combination
  - Feature augmentation
- Parallelized
  - Mixed
  - Weighted
  - Switching
- Pipelined
  - Cascade
  - Meta-level

Parallelized hybridization

One data source

- Multiple recommenders
- Combined output



Parallelized hybridization

Least invasive design

- Recommenders as black boxes

Hybridization by data fusion

- Some weighting or voting scheme
- Weights can be learned dynamically
- Switching in the extreme case

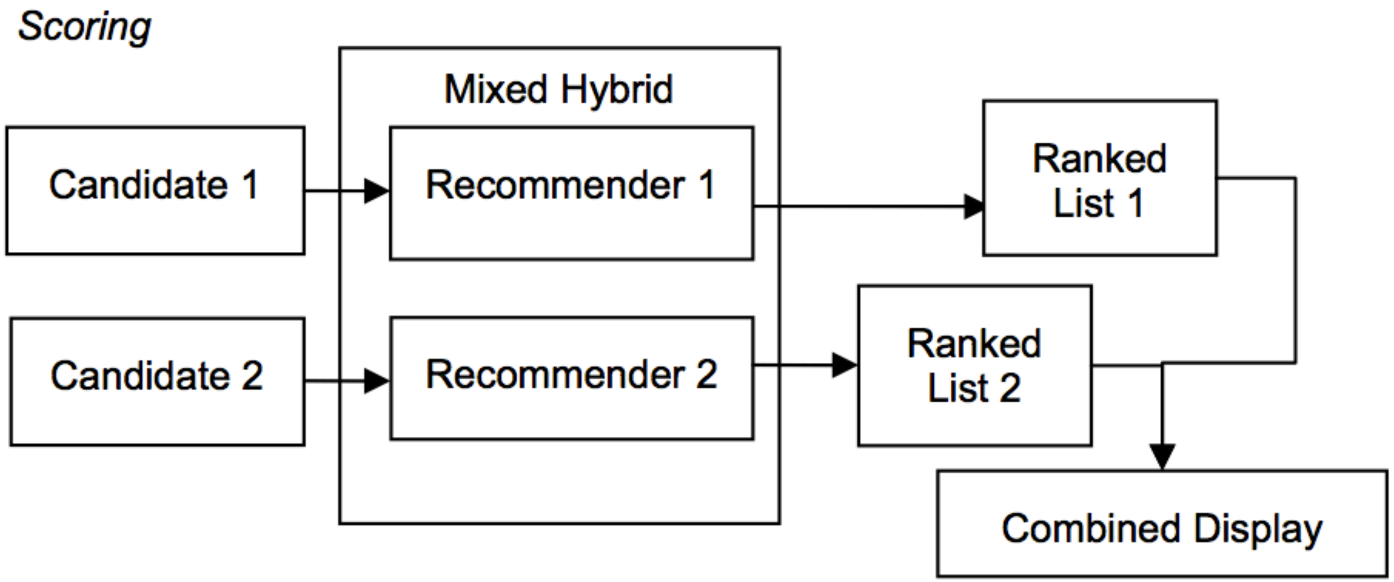
Mixed hybridization

Parallelized

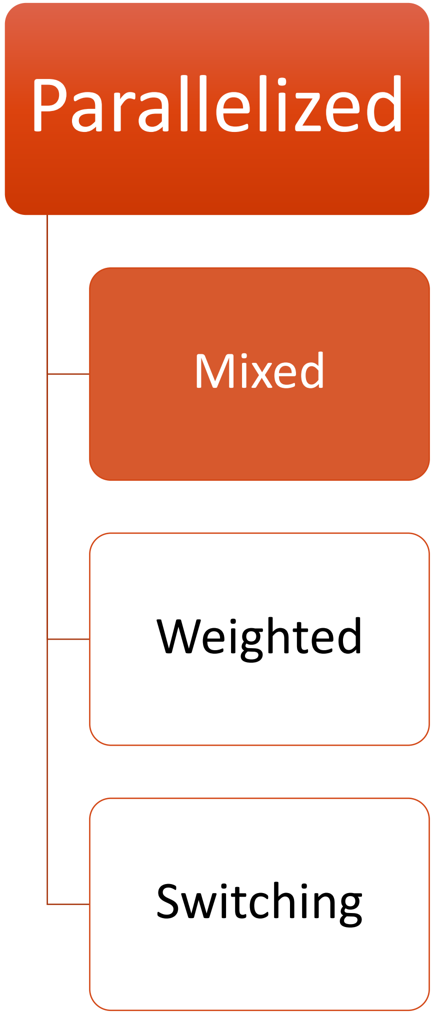
- Mixed
- Weighted
- Switching

Recommendations put together

- Results combined at the interface level



Mixed hybridization



CB/CF TV show recommender [Smyth and Cotter, KBS 2000]

- CB recommendations
- Followed by CF recommendations

Tourist bundles recommender [Zanker et al., WISE 2007]

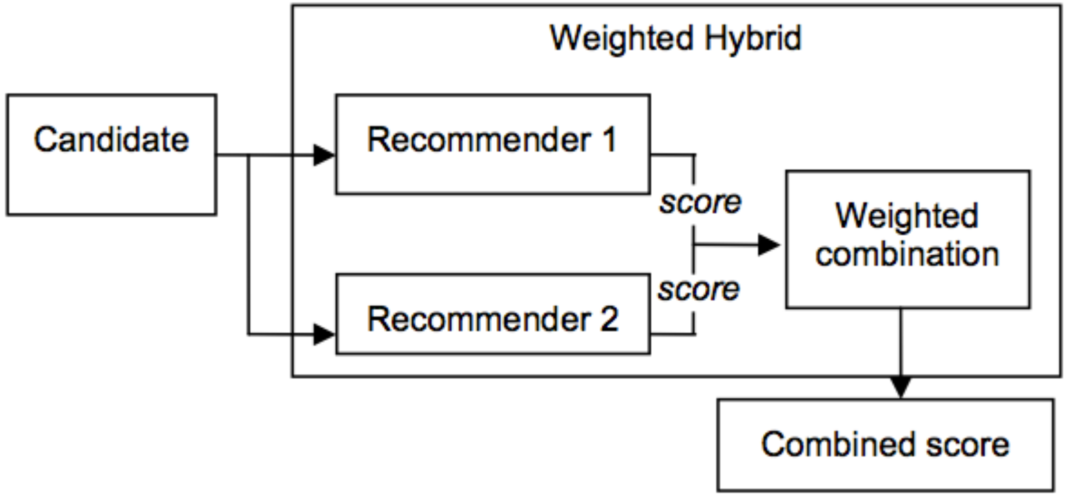
- Category-specific recommenders
  - e.g., accommodation, leisure activities
- One recommendation per category
- Hybridization as a constraint satisfaction problem

Weighted hybridization

- Parallelized
  - Mixed
  - Weighted
  - Switching

Scores linearly combined with weight vector $\vec{\boldsymbol{w}}$

- $\hat{r}_{ui} = \sum_{j=1}^{k} w_j \, \hat{r}_{ui}^{(j)}$
  - w_j: weight of j-th recommender
  - $\hat{r}_{ui}^{(j)}$: prediction by j-th recommender



Weighted hybridization

- Parallelized
  - Mixed
  - Weighted
  - Switching

Scores linearly combined with weight vector $\vec{\boldsymbol{w}}$

- $\hat{r}_{ui} = \sum_{j=1}^{k} w_j \, \hat{r}_{ui}^{(j)}$
  - w_j: weight of j-th recommender
  - $\hat{r}_{ui}^{(j)}$: prediction by j-th recommender

$\boldsymbol{w_1}$	$\boldsymbol{w_2}$		$\hat{r}_{ui}^{(1)}$	$\hat{r}_{ui}^{(2)}$	$\hat{r}_{ui}$
0.5	0.5	i_1	0.5	0.8	0.65
		i_2	0.1	0.0	0.05

Weighted hybridization

- **Parallelized**
  - Mixed
  - **Weighted**
  - Switching

How to optimize weights?

- Minimize error on training data

w_1	w_2		$\hat{r}_{ui}^{(1)}$	$\hat{r}_{ui}^{(2)}$	$\hat{r}_{ui}$	Error	MAE
0.1	0.9	i_1	0.5	0.8	0.77	0.23	0.61
		i_2	0.1	0.0	0.01	0.99	
0.3	0.7	i_1	0.5	0.8	0.71	0.29	0.63
		i_2	0.1	0.0	0.03	0.97	
0.5	0.5	i_1	0.5	0.8	0.65	0.35	0.65
		i_2	0.1	0.0	0.05	0.95	
0.7	0.3	i_1	0.5	0.8	0.59	0.41	0.67
		i_2	0.1	0.0	0.07	0.93	
0.9	0.1	i_1	0.5	0.8	0.53	0.47	0.69
		i_2	0.1	0.0	0.09	0.91	

minimum error

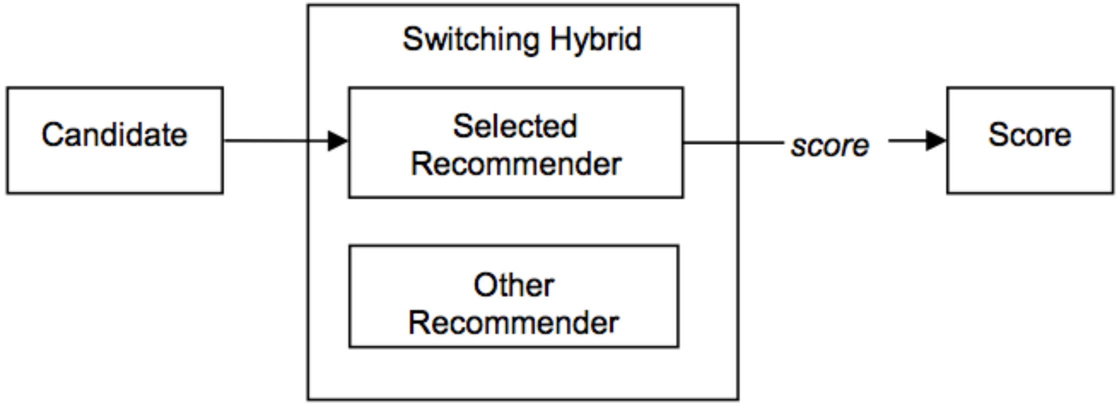
optimal weights

Switching hybridization

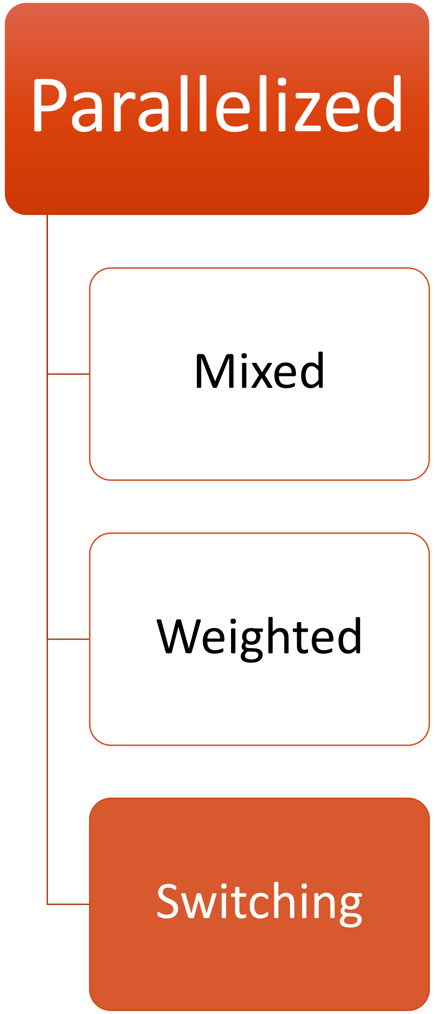
- **Parallelized**
  - Mixed
  - Weighted
  - **Switching**

Extreme case of weighted hybrids

- $\exists_1 \, j \in 1 \ldots k \mid \hat{r}_{ui} = \hat{r}_{ui}^{(j)}$



Switching hybridization



Extreme case of weighted hybrids

- $\exists_1\, j \in 1 \ldots k \mid \hat{r}_{ui} = \hat{r}_{ui}^{(j)}$

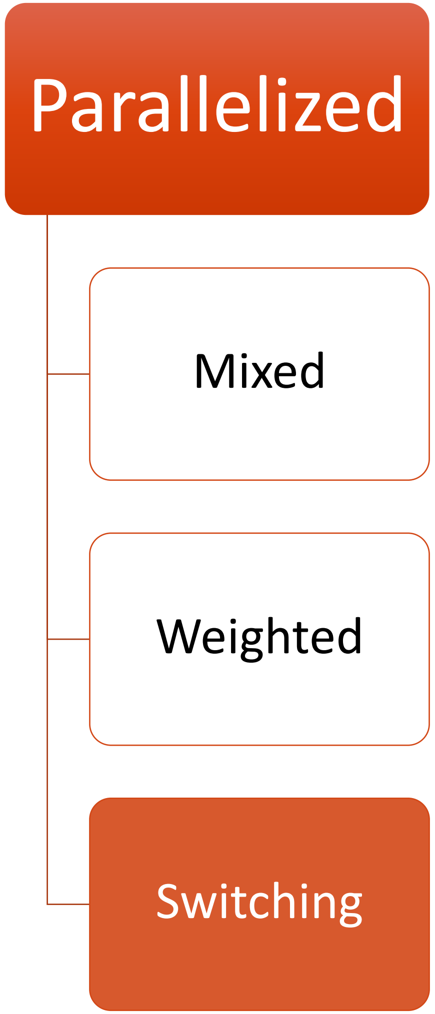
Quality-based decision

- Too few ratings? Content-based
- Else? Collaborative filtering

Learned on a case-by-case basis

- Akin to adaptive approaches

Switching hybridization



CB/CF/CB news recommendation
[Billsus and Pazzani, UMUAI 2000]

- recs = CB-kNN(•)
  not enough confidence?
    recs = CF(•)
  not enough confidence?
    recs = CB-Naive(•)
  return recs

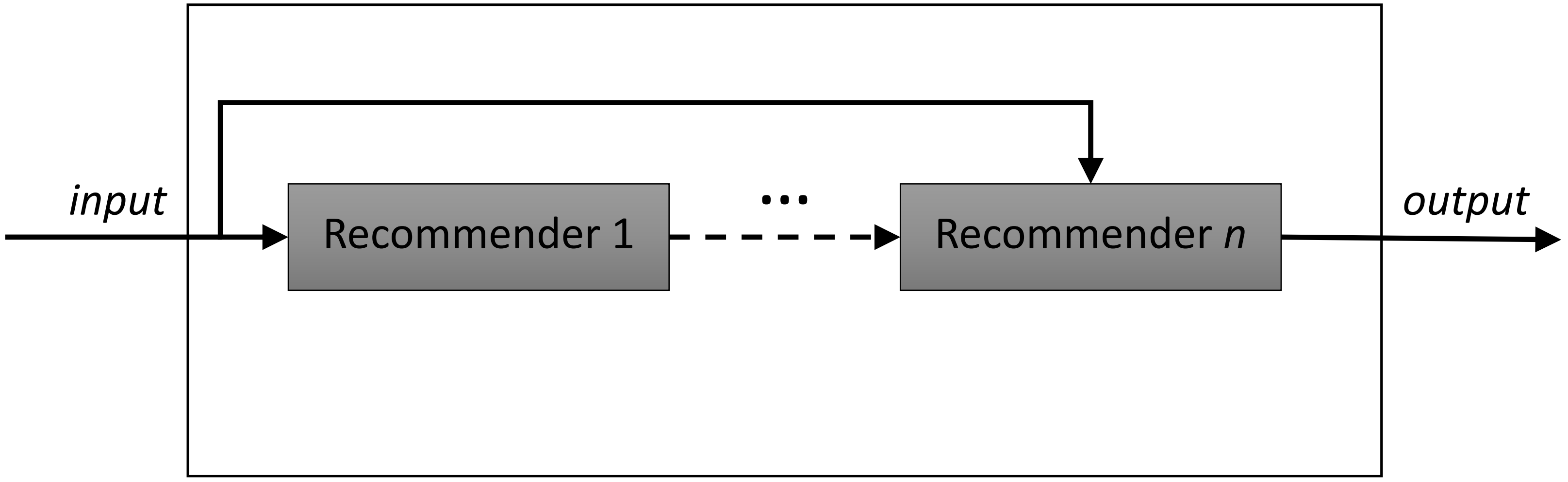
Hybridization designs [Burke, UMUAI 2002]

- **Monolithic**
  - Feature combination
  - Feature augmentation
- **Parallelized**
  - Mixed
  - Weighted
  - Switching
- **Pipelined**
  - Cascade
  - Meta-level

Pipelined hybridization

One data source

- Multiple recommenders
- Sequential refinement

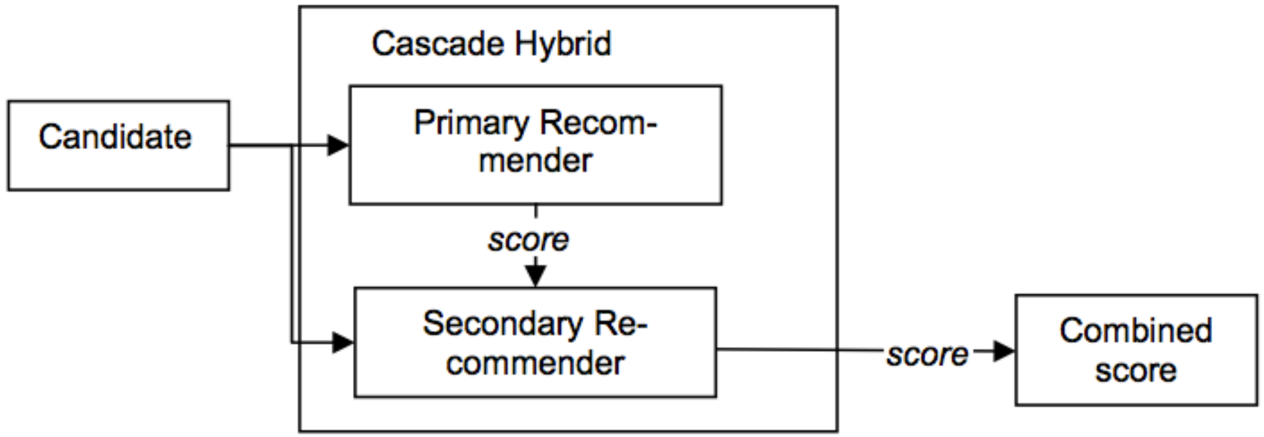


Cascade hybridization

- Pipelined
  - Cascade
  - Meta-level

Recommendations restricted by predecessor

- $\hat{r}_{ui} = \hat{r}_{ui}^{(k)}$ and
- $\forall j \in 1 \dots k,\ \hat{r}_{ui}^{(j)} = \begin{cases} \hat{r}_{ui}^{(j)} & \hat{r}_{ui}^{(j-1)} > 0 \\ 0 & \text{otherwise} \end{cases}$



Cascade hybridization

- Pipelined
  - Cascade
  - Meta-level

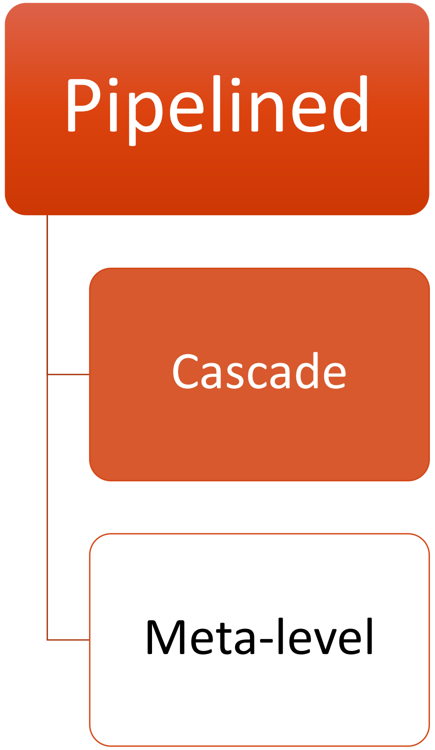
Recommendations restricted by predecessor

- $\hat{r}_{ui} = \hat{r}_{ui}^{(k)}$ and
- $\forall j \in 1 \dots k,\ \hat{r}_{ui}^{(j)} = \begin{cases} \hat{r}_{ui}^{(j)} & \hat{r}_{ui}^{(j-1)} > 0 \\ 0 & \text{otherwise} \end{cases}$

Subsequent recommender may not introduce items

- Produces very precise results

Cascade hybridization



KB/CF restaurant recommendation [Burke, UMUAI 2002]

- KB recommendation
  - Keep matching restaurants
- CF recommendation
  - Break ties

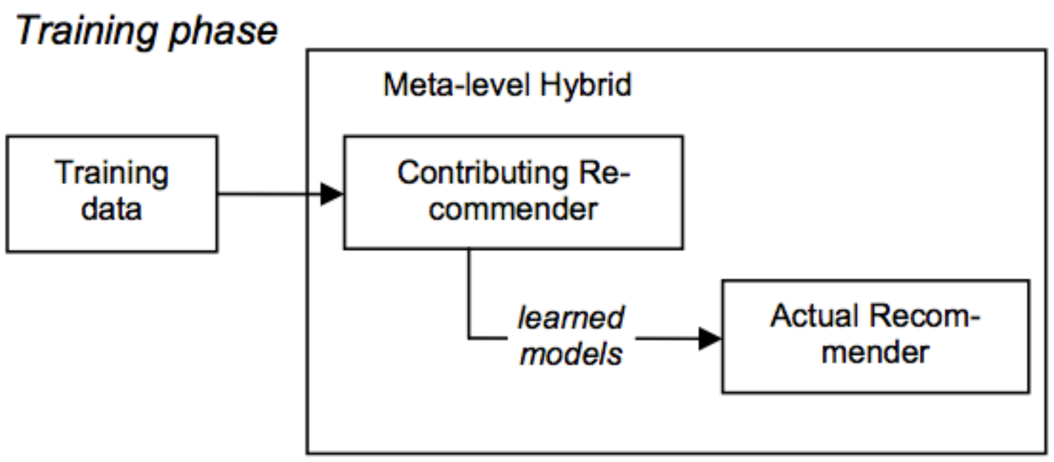
Meta-level hybridization

Pipelined

- Cascade
- Meta-level

Recommender based on predecessor's ***models***

- Similar to feature augmentation, except that actual recommender doesn't work with raw data



Meta-level hybridization

- Pipelined
  - Cascade
  - Meta-level

CB/CF restaurant recommendation [Pazzani, AI Review 1999]

- CB user modeling
  - Naïve Bayes classification
- CF recommendation
  - User-based CF

Summary

- Monolithic
  - Feature combination
  - Feature augmentation
- Parallelized
  - Mixed
  - Weighted
  - Switching
- Pipelined
  - Cascade
  - Meta-level

Summary

Lack of empirical comparison of designs

- Multiple paradigms rarely supported in one dataset

Few conclusions supported by empirical findings

- Monolithic: preprocessing effort traded for knowledge
- Parallel: requires careful matching of scores
- Pipelined: works well for antithetic approaches

Summary

Netflix competition winner

- Weighted design based on 100+ recommenders
- Switching of weights given user and context model

References

[Recommender Systems: An Introduction] (Ch. 5)

[Recommender Systems: The Textbook] (Ch. 6)