ICDE2023
Anaheim, CA April 3 – 7, 2023

MAMDR: A Model Agnostic Learning Framework for Multi-Domain Recommendation

Linhao Luo, Yumeng Li, Buyu Gao, Shuai Tang, Sinan Wang, Jiancheng Li, Tanchao Zhu, Jiancai Liu, Zhao Li, and Shirui Pan

Monash University, Alibaba Group, Hangzhou Link2Do Technology, and Griffith University

Presenter: Linhao Luo

Background

Recommender System

Recommender systems have been widely applied in many applications.

淘宝网 Taobao.com

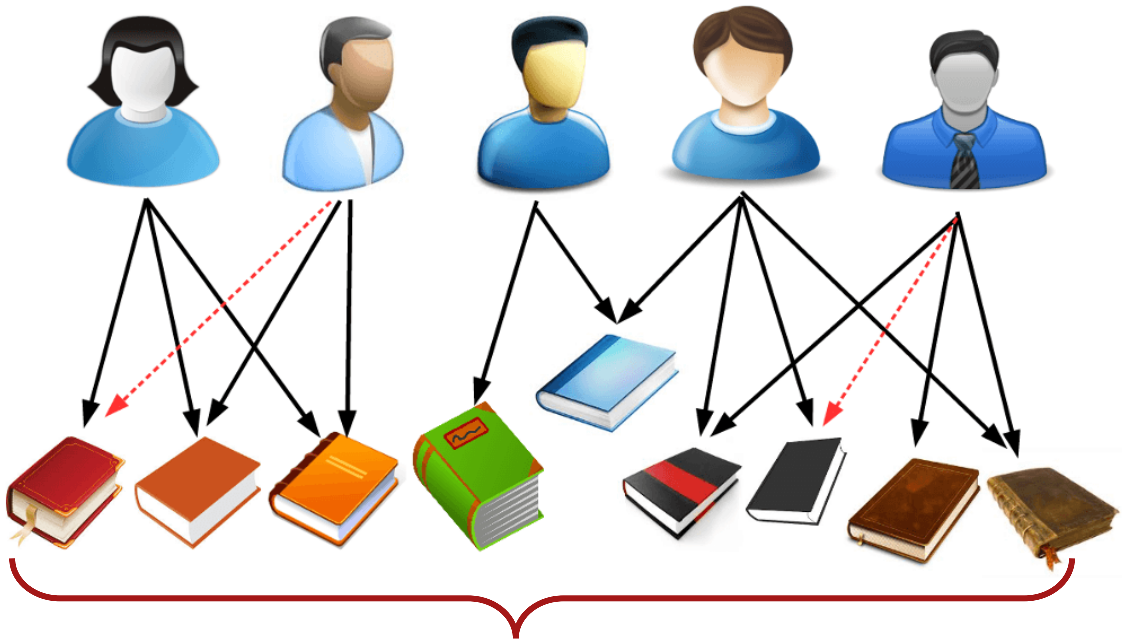
NETFLIX

amazon.com

Spotify

Single-domain Recommendation

Example: User-item recommender system.



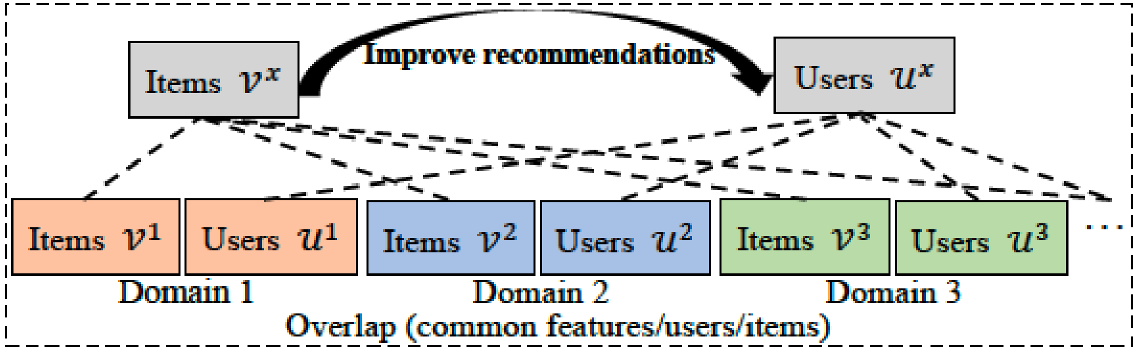
Conventional recommender systems are trained and predicted on samples collected from a **single domain.**

Multi-domain Recommendation (MDR)

Cosmetics

Outdoor exercises

House cleanings

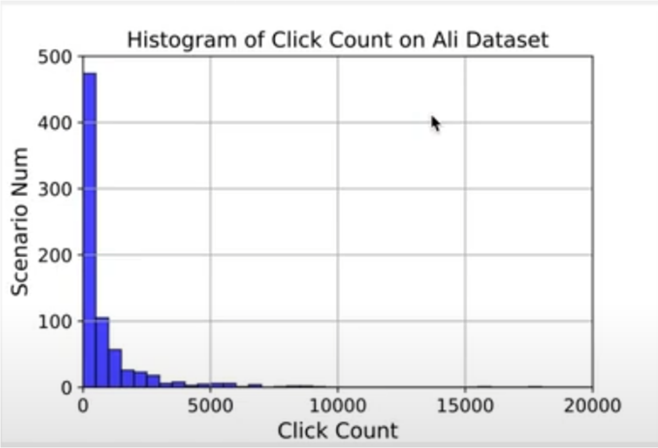


Multi-domain recommendation

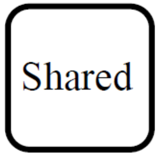
Challenges

- Long-tailed data distribution across domains.

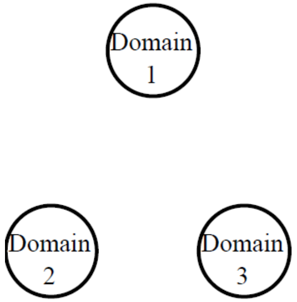
Singles Day Promotion (Double 11)



Long-tailed data distribution



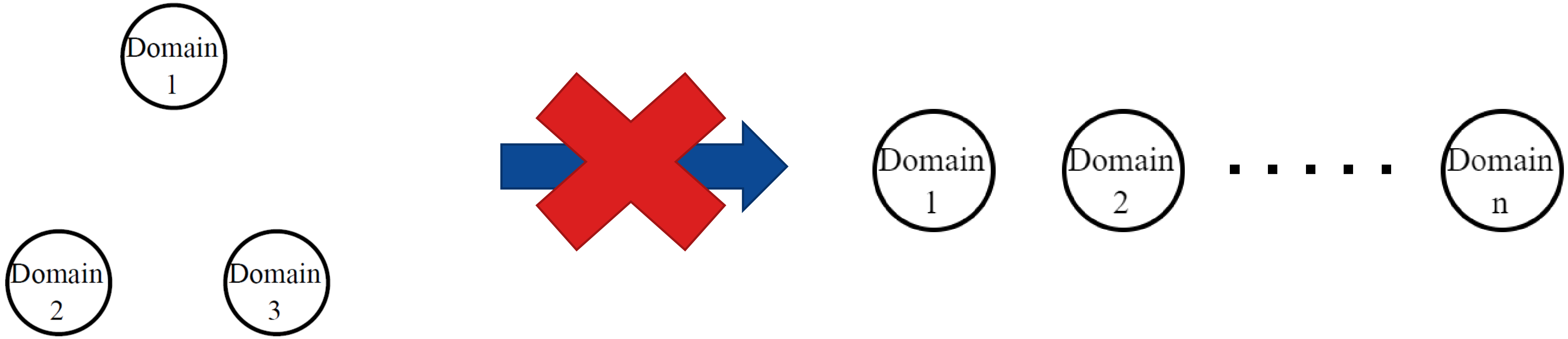
One shared model for all domains



One model for each domain

Challenges

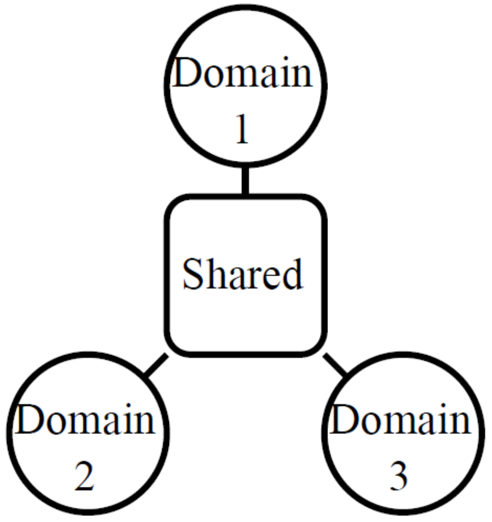
- Designing a model for each domain is time-consuming and unscalable.



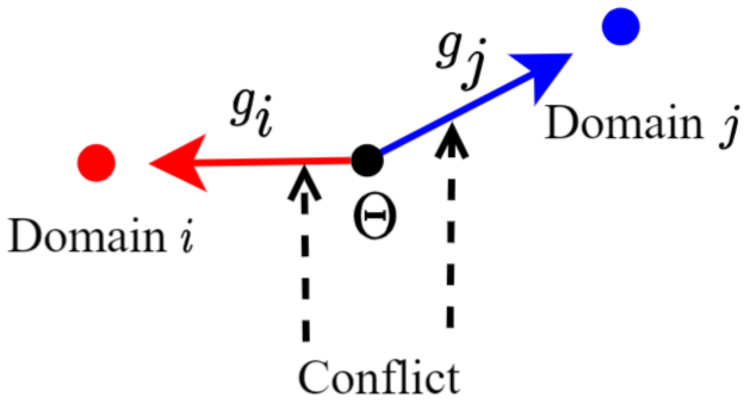
One model for each domain **Thousands of domains**

Challenges

- Existing MDR methods cannot generalize to all circumstances.
- Shared parameters suffer from the **domain conflict**, and specific parameters are inclined to **overfitting**.



One model with shared and specific parameters



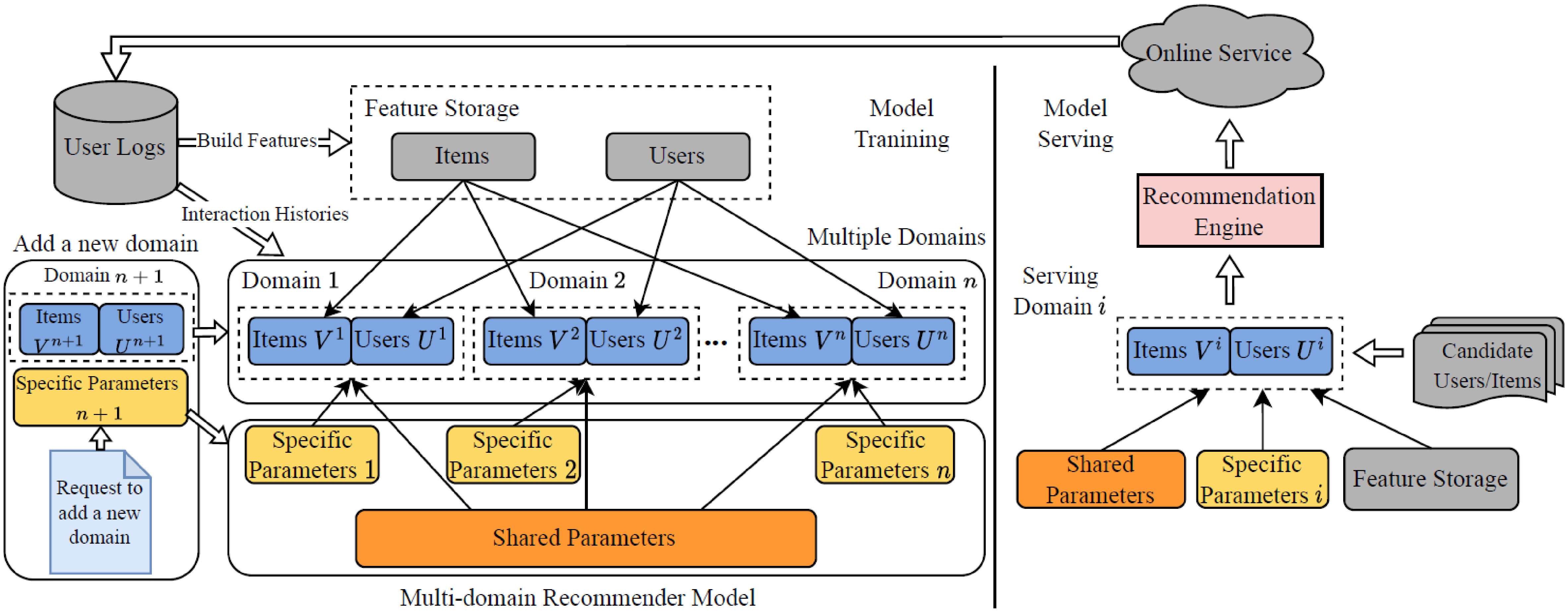
Domain Conflict

Contributions

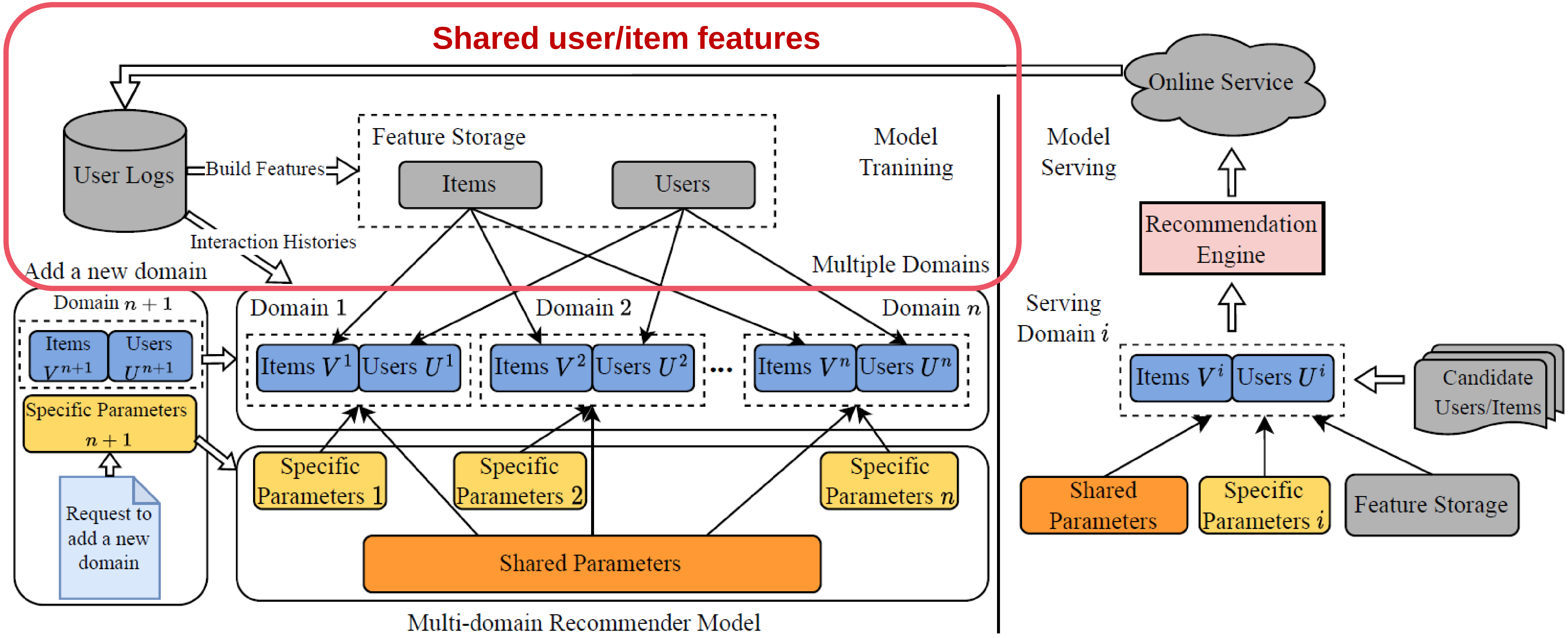
- We present a **multi-domain recommender system** served in Taobao, and propose a novel **model agnostic** learning framework: **MAMDR**, which is compatible with **arbitrary model structures**.
- We propose two scalable algorithms: **Domain Negotiation (DN)** and **Domain Regularization (DR)** to alleviate the domain conflict and overfitting problem in MDR.
- We provide a **disturbed implementation** of MAMDR and publicized various benchmark **datasets** to simulate the real-world challenges in MDR problems.

Methodology

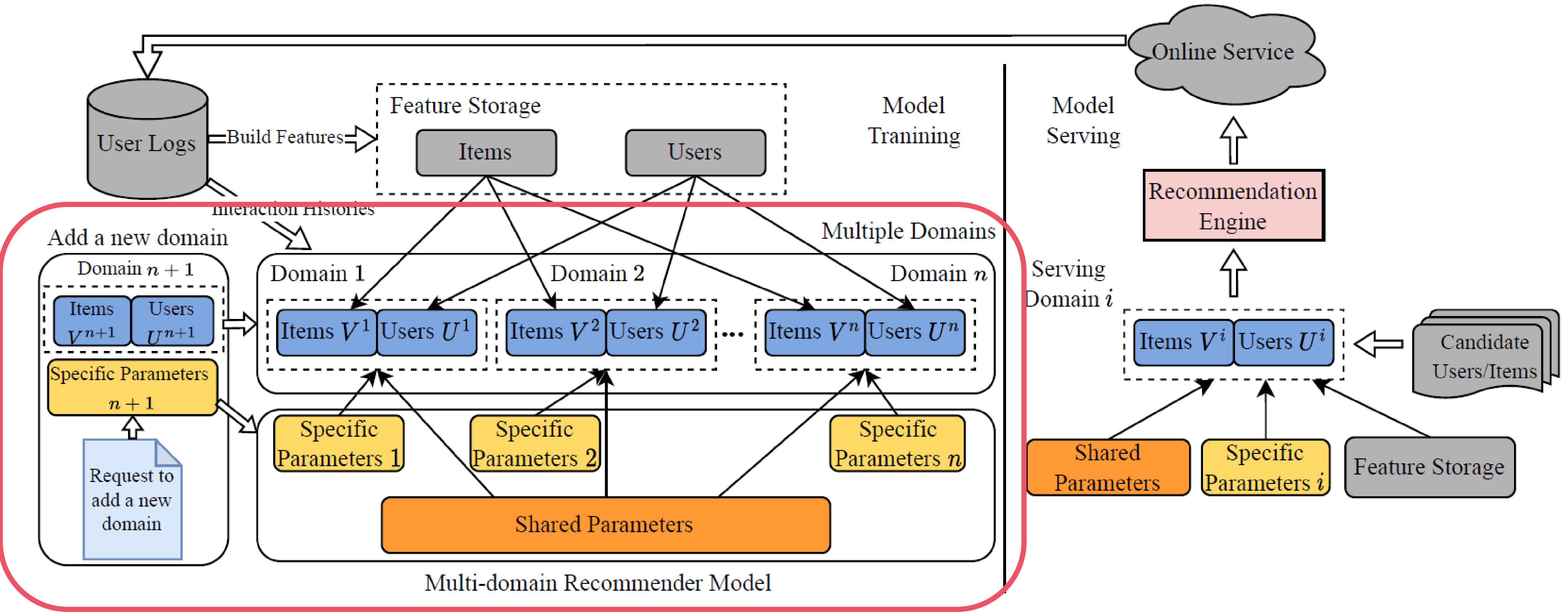
Multi-Domain Recommender system served in Taobao



Multi-Domain Recommender system served in Taobao

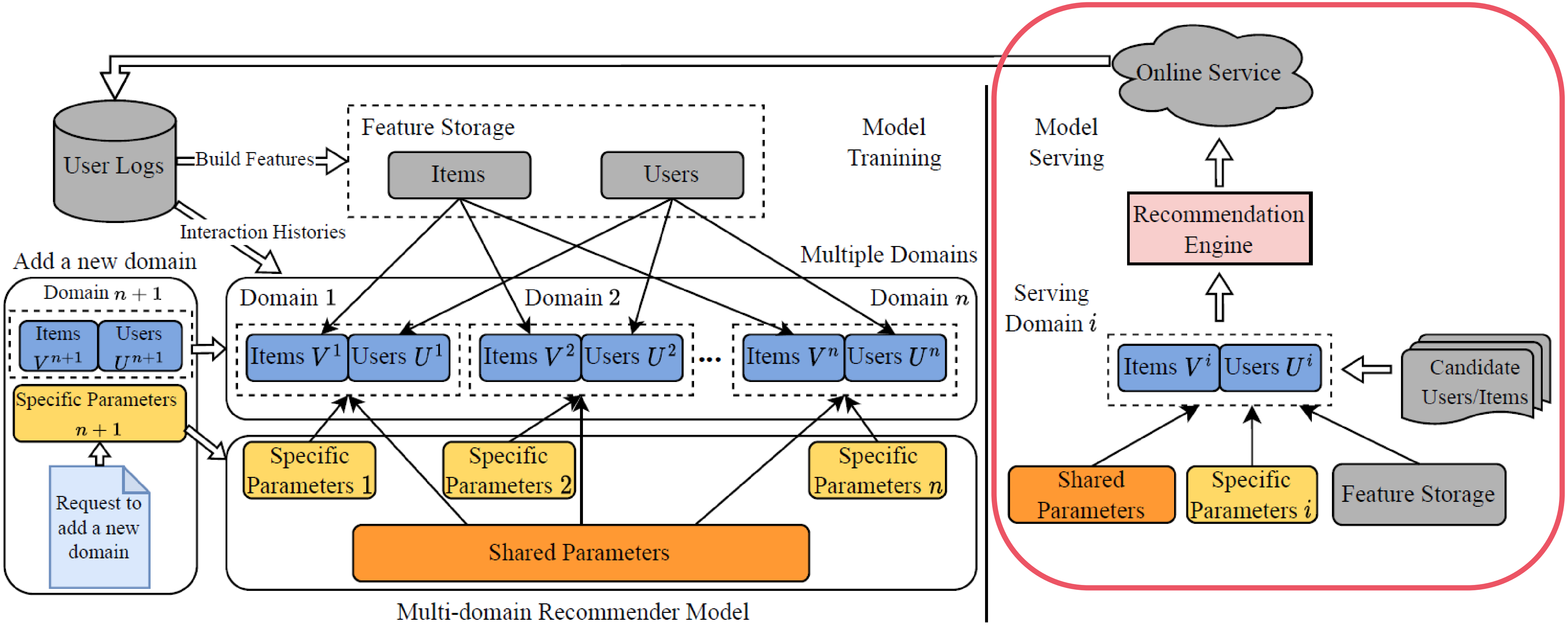


Multi-Domain Recommender system served in Taobao



Scale to new domain by increasing specific parameters.

Multi-Domain Recommender system served in Taobao



How to combine shared and specific parameters for serving?

Domain Negotiation (DN)

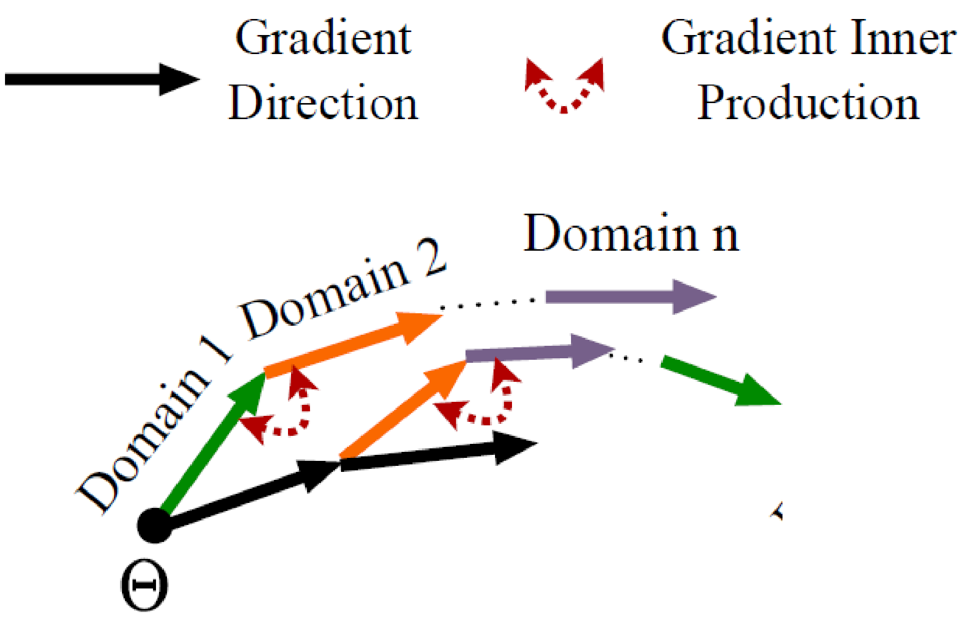
- Domain Negotiation (DN) is proposed to mitigate the domain conflict problem for **shared parameters.**

Algorithm 1: Domain Negotiation (DN)

Input: n different domains $\mathcal{D}$, initial model parameters Θ, learning rate α and β, maximum training epoch N.

Output: Θ

for $epoch = 1, \cdots, N$ **do**
    $\widetilde{\Theta}_1 \leftarrow \Theta$;
    Randomly shuffle $\mathcal{D}$;
    for $i \leftarrow 1, \cdots, n$ **do**
        Update $\widetilde{\Theta}_{i+1} \leftarrow \widetilde{\Theta}_i - \alpha \cdot \nabla L(\widetilde{\Theta}_i, T^i)$;
    end
    Update $\Theta \leftarrow \Theta + \beta \cdot (\widetilde{\Theta}_{n+1} - \Theta)$;
end
return Θ



a. Domain Negotiation

Analysis of Domain Negotiation (DN)

In DN, we serially perform training on each domain, which provides a sequence of loss $L(\widetilde{\Theta}_i, T^i)$, which can be simplified as $L_i(\widetilde{\Theta}_i)$. The following notations are also defined to facilitate analysis.

$$g_i = L_i'(\widetilde{\Theta}_i) \qquad \text{(gradients from domain } i\text{)}, \tag{4}$$

$$\overline{g}_i = L_i'(\widetilde{\Theta}_1) \qquad \text{(gradients from domain } i \text{ at initial point)}, \tag{5}$$

$$\overline{H}_i = L_i''(\widetilde{\Theta}_1) \qquad \text{(Hessian at initial point)}, \tag{6}$$

$$\widetilde{\Theta}_i = \widetilde{\Theta}_1 - \alpha \sum^{i-1} g_j \qquad \text{(sequence of gradient descent)}. \tag{7}$$

Analysis of Domain Negotiation (DN)

We can perform the Taylor expansion on the g_i when α is small enough, which is formulated as:

$$
\begin{aligned}
g_i &= L_i'(\widetilde{\Theta}_1) + L''(\widetilde{\Theta}_1)(\widetilde{\Theta}_i - \widetilde{\Theta}_1) + O(\alpha^2), && (14)\\
&= \overline{g}_i + \overline{H}_i(\widetilde{\Theta}_i - \widetilde{\Theta}_1) + O(\alpha^2), && (15)\\
&= \overline{g}_i - \alpha\overline{H}_i \sum_{j=1}^{i-1} g_j + O(\alpha^2), && (16)\\
&= \overline{g}_i - \alpha\overline{H}_i \sum_{j=1}^{i-1} \overline{g}_j + O(\alpha^2) \quad (\text{using } g_j = \overline{g}_j + O(\alpha)). && (17)
\end{aligned}
$$

Analysis of Domain Negotiation (DN)

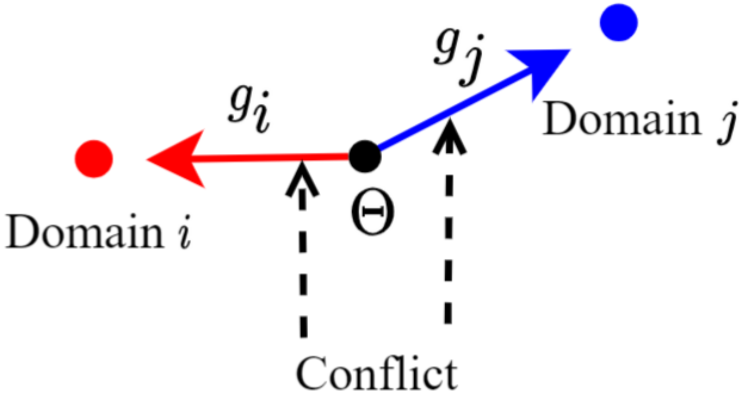
Then, the gradients $\widetilde{\Theta}_{n+1} - \Theta$ for outer loop can be formulated as:

$$-(\widetilde{\Theta}_{n+1} - \Theta)/\alpha = \sum_{i=1}^{n} g_j = \boxed{\sum_{i=1}^{n} \overline{g}_i} - \boxed{\alpha \sum_{i=1}^{n} \sum_{j=1}^{i-1} \overline{H}_i \overline{g}_j} + O(\alpha^2). \quad (18)$$

$$\text{InnerGrad} = \mathbb{E}(\overline{H}_i \overline{g}_j) = \mathbb{E}(\overline{H}_j \overline{g}_i), \quad (19)$$

$$= \frac{1}{2} \mathbb{E}(\overline{H}_i \overline{g}_j + \overline{H}_j \overline{g}_i), \quad (20)$$

$$= \frac{1}{2} \mathbb{E}\left(\frac{\partial}{\partial \Theta} \langle \overline{g}_i, \overline{g}_j \rangle\right). \quad (21)$$



Domain Conflict

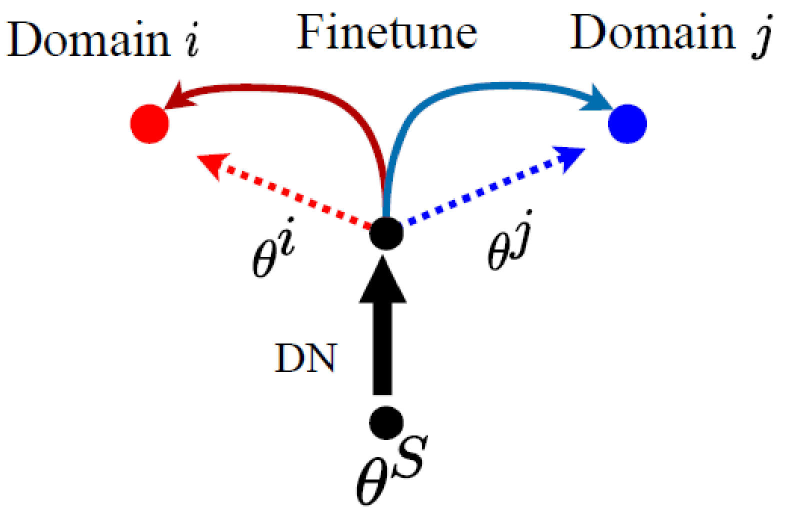
Domain Regularization (DR)

- Domain-specific parameters θ^i.

$$\Theta = \theta^S + \theta^i$$

θ^i can be treated as a direction pointing to the optimal position of each domain.

However, it is easy to overfit on some data sparsity domains.



a. Specific Parameters and Finetune

Domain Regularization (DR)

- Domain Regularization (DR) utilize data from other domains to help training specific parameters

Algorithm 2: Domain Regularization (DR)

Input: n different domains $\mathcal{D}$, target domain D^i, specific parameters θ^i, learning rate α, γ, sample number k

Output: θ^i

Sample k domains from $\mathcal{D}$ as $\widetilde{\mathcal{D}}$;

for D^j *in* $\widetilde{\mathcal{D}}$ **do**

  $\widetilde{\theta}^i \leftarrow \theta^i$;

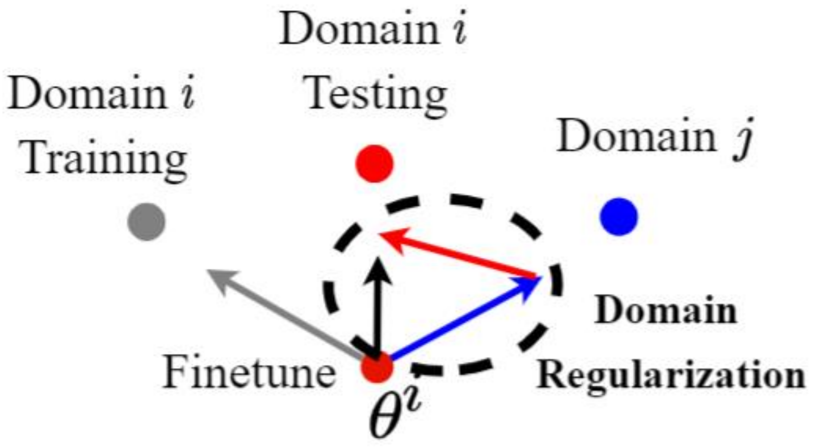
  Update $\widetilde{\theta}^i \leftarrow \widetilde{\theta}^i - \alpha \cdot \bigtriangledown L(\widetilde{\theta}^i, T^j)$ # Update on domain j;

  Update $\widetilde{\theta}^i \leftarrow \widetilde{\theta}^i - \alpha \cdot \bigtriangledown L(\widetilde{\theta}^i, T^i)$ # Using domain i as regularization;

  Update $\theta^i \leftarrow \theta^i + \gamma \cdot (\widetilde{\theta}^i - \theta^i)$;

end

return θ^i



b. Domain Regularization

$$-(\widetilde{\theta}^i - \theta^i)/\alpha = g_j + g_i = \overline{g}_j + \overline{g}_i - \boxed{\alpha \overline{H}_i \overline{g}_j}$$

This regularizes the gradients on domain j to optimize domain i.

Model Agnostic Learning Framework for Multi-Domain Recommendation

Algorithm 3: MAMDR

Input: n different domains $\mathcal{D}$, shared parameters θ^S, domain-specific parameters $\{\theta^1, \cdots, \theta^n\}$, learning rate α, β, γ, sample size k, maximum training epoch N.

Output: $\Theta = \{\theta^S, \{\theta^1, \cdots, \theta^n\}\}$

for $epoch = 1, \cdots, N$ **do**
    Update θ^S using Domain Negotiation (Algorithm 1);
    for $i = 1, \cdots, n$ **do**
        Update θ^i using Domain Regularization (Algorithm 2);
    end
end
return $\Theta = \{\theta^S, \{\theta^1, \cdots, \theta^n\}\}$

MAMDR is **agnostic to model structure** that can be incorporated with any existing recommendation model.

Large-scale Implementation

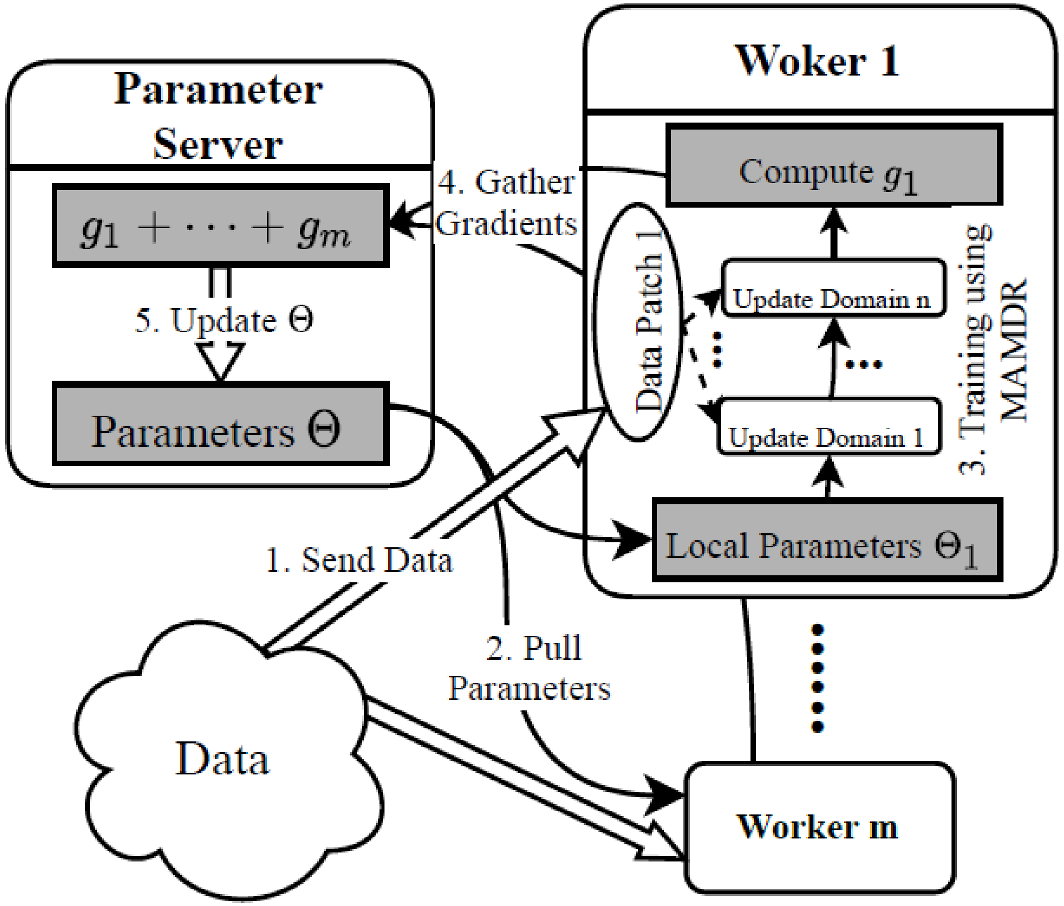


Fig. 6. Implementation of MAMDR in PS-Worker architecture.

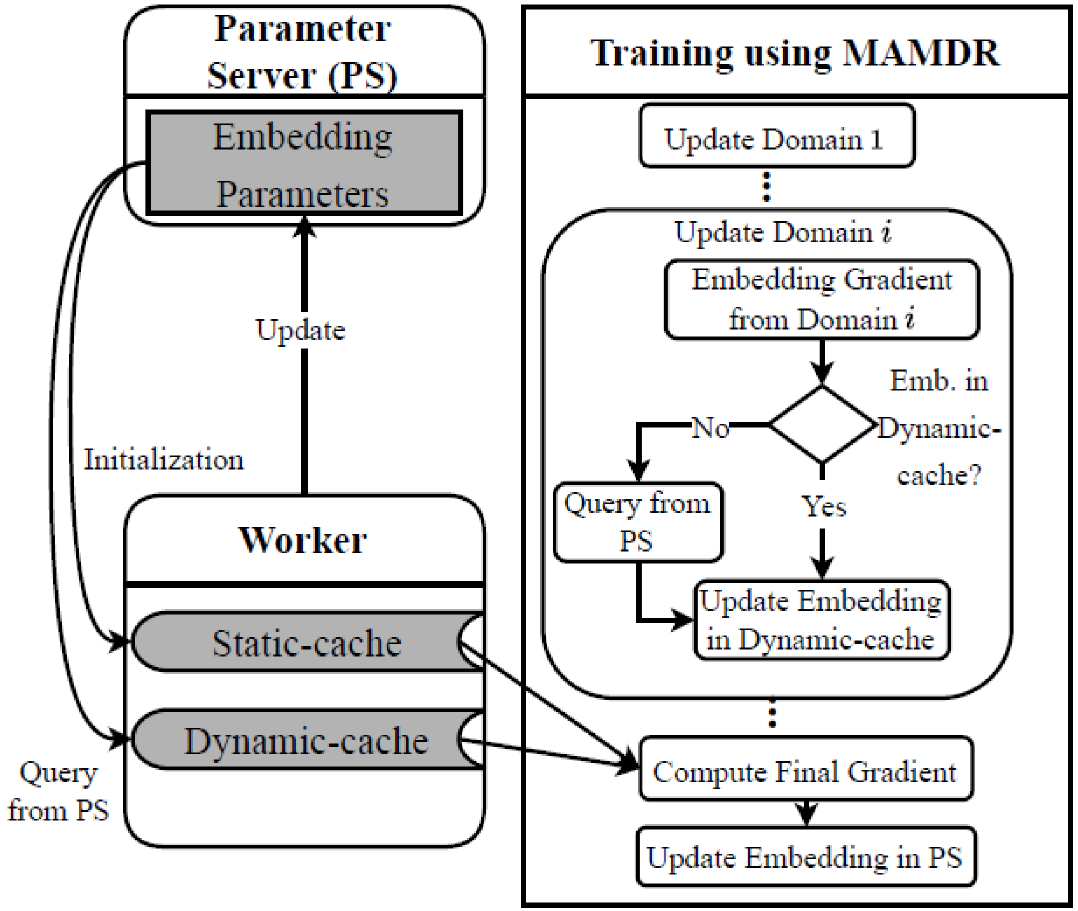


Fig. 7. The illustration of Embedding PS-Worker cache.

Experiments

Dataset

TABLE I

OVERALL STATISTIC OF DATASETS.

Dataset	#Domain	#User	#Item	#Train	#Val	#Test	Sample / Domain
Amazon-6	6	445,789	172,653	9,968,333	3,372,666	3,585,877	2,821,146
Amazon-13	13	502,222	215,403	11,999,607	4,100,756	4,339,523	1,572,299
Taobao-10	10	23,778	6,932	92,137	37,645	43,502	17,328
Taobao-20	20	58,190	16,319	243,592	96,591	106,500	22,334
Taobao-30	30	99,143	29,945	394,805	151,369	179,252	24,180
Taobao-online	69,102	84,307,785	16,385,662	420,097,203	23,340,352	46,415,298	7,088

Metrics: AUC

Performance comparison

- Performance on MDR benchmark datasets

TABLE V
COMPARISON WITH MULTI-DOMAIN RECOMMENDATION METHODS UNDER AVERAGE AUC AND AVERAGE RANK METRICS.

	Method	Amazon-6		Amazon-13		Taobao-10		Taobao-20		Taobao-30	
		AUC	RANK	AUC	RANK	AUC	RANK	AUC	RANK	AUC	RANK
Single domain	MLP	0.7464	9.0	0.7016	8.6	0.7022	11.3	0.7255	9.9	0.7416	10.7
	WDL	0.7449	9.0	0.7026	7.9	0.7154	8.9	0.7235	10.6	0.7559	8.4
	NeurFM	0.6505	10.7	0.6152	10.2	0.7374	4.1	0.7461	6.4	0.7673	6.1
	AutoInt	0.7531	8.2	0.7214	6.4	0.7302	5.8	0.7471	6.3	0.7623	6.5
	DeepFM	0.7333	10.0	0.6976	8.5	0.7271	6.6	0.7347	8.8	0.7484	9.4
Multi domain	Shared-bottom	0.7794	3.0	0.7088	5.0	0.7197	7.7	0.7572	4.3	0.7714	6.1
	MMOE	0.7816	2.7	0.7381	4.2	0.7250	5.9	0.7494	6.0	0.7717	4.2
	PLE	0.7801	3.5	0.7114	6.3	0.7287	5.3	0.7603	3.3	0.7725	4.0
	Star	0.7719	5.8	0.7209	7.1	0.7202	8.0	0.7324	8.9	0.7483	9.4
	MLP+MAMDR	**0.7957**	**2.5**	**0.7577**	**3.5**	**0.7445**	**2.7**	**0.7613**	**3.2**	**0.7750**	**3.1**

Performance comparison

- Performance on industry datasets

TABLE VIII
RESULTS ON THE INDUSTRY DATASET UNDER AVERAGE AUC METRIC.

Methods	RAW	MMOE	CGC	PLE	RAW+Separate	RAW+DN	RAW+MAMDR
AUC	0.7503	0.7497	0.7489	0.7513	0.7460	0.7559	**0.7700**

Average AUC of 69,102 domains

TABLE IX
RESULTS ON TOP 10 LARGEST DOMAINS OF INDUSTRY DATASET UNDER AUC METRIC.

Methods	Top 1	Top 2	Top 3	Top 4	Top 5	Top 6	Top 7	Top 8	Top 9	Top 10
RAW	0.8202	0.7635	0.8439	0.7295	0.6962	0.7417	0.6661	0.7524	0.7540	0.6912
MMOE	0.8166	0.7597	0.8288	0.7694	0.6945	0.7453	0.6677	0.7315	0.7478	0.6941
CGC	0.8172	0.7640	0.8307	0.7747	0.7215	0.7392	0.6726	0.7444	0.7357	0.7019
PLE	0.8158	0.7643	0.8261	0.7768	0.7327	0.7284	0.6793	0.7410	0.7472	0.7038
RAW+Separate	0.8127	0.7635	0.8285	0.7569	0.6896	0.7367	0.6701	0.7370	0.7283	0.6947
RAW+DN	0.8173	0.7655	0.8397	0.7643	0.7188	0.7344	0.6664	0.7523	0.7505	0.7021
RAW+MAMDR	**0.8226**	**0.7704**	**0.8469**	**0.8090**	**0.7391**	**0.7648**	**0.6965**	**0.7666**	**0.7689**	**0.7150**

Learning Framework Comparison

TABLE X

COMPARISON WITH OTHER LEARNING FRAMEWORKS UNDER AVERAGE AUC METRIC ON TAOBAO-10.

Method	Alternate	Alternate+Finetune	Weighted Loss	PCGrad	MAML	Reptile	MLDG	DN	DR	MAMDR (DN+DR)
MLP	0.7022	0.7126	0.7157	0.7254	0.6896	0.7117	0.7074	0.7204	0.7407	**0.7445**
WDL	0.7154	0.7040	0.7098	0.7153	0.6945	0.7212	0.7182	0.7295	0.7346	**0.7376**
NeurFM	0.7154	0.7465	0.7393	0.7526	0.7479	0.7579	0.7543	0.7572	0.7553	**0.7609**
DeepFM	0.7271	0.7280	0.7259	0.7562	0.7237	0.7402	0.7480	0.7352	0.7466	**0.7581**
Shared-bottom	0.7197	0.7225	0.7171	0.7269	0.6816	0.7255	0.7195	0.7233	0.7244	**0.7339**
Star	0.7202	0.7303	0.7297	0.7221	0.7228	0.7353	0.7181	0.7328	0.7255	**0.7520**

Traditional Learning Framework **Multi-Task Learning Framework** **Meta-Learning Framework**

Parameters Analysis

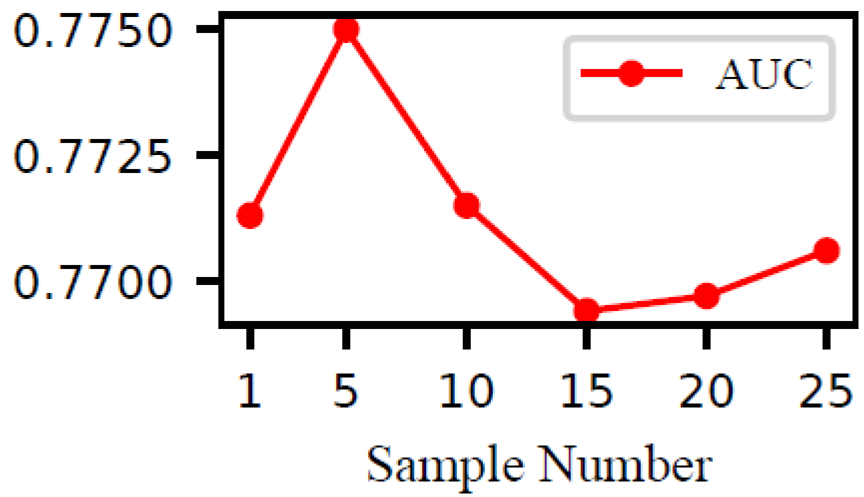


Fig. 8. Results under different sample number k.

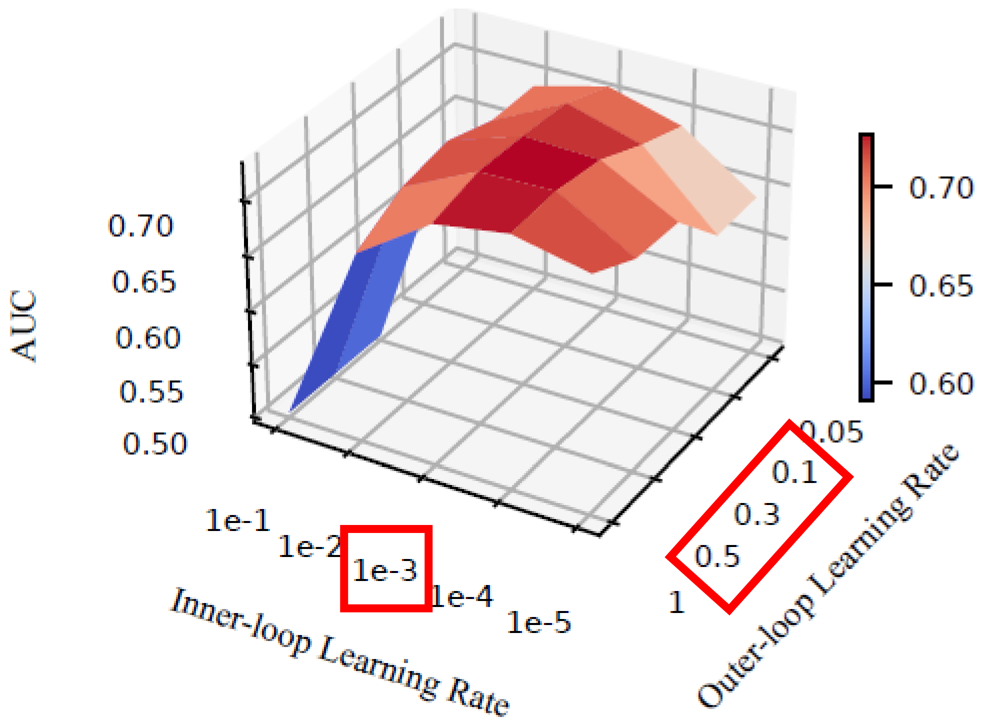


Fig. 9. Results under different learning rates.

Thanks for your listening!

Code & Data