[SIGIR20]

How to Retrain Recommender System? A Sequential Meta-Learning Method*

Yang Zhang[1], Fuli Feng[2], Chenxu Wang[1], Xiangnan He[1], Meng Wang[3], Yan Li[4], Yongdong Zhang[1]
[1]University of Science and Technology of China, [2]National University of Singapore
[3]Hefei University of Technology, [4]Beijing Kuaishou Technology Co., Ltd. Beijing, China
{fulifeng93,xiangnanhe}@gmail.com,{zy2015,wcx123}@mail.ustc.edu.cn
eric.mengwang@gmail.com,liyan@kuaishou.com,zhyd73@ustc.edu.cn

[SIGIR20]*How to Retrain Recommender System? A Sequential Meta-Learning Method*

- Fine-tuning
- Sample-based retraining
- Full retraining

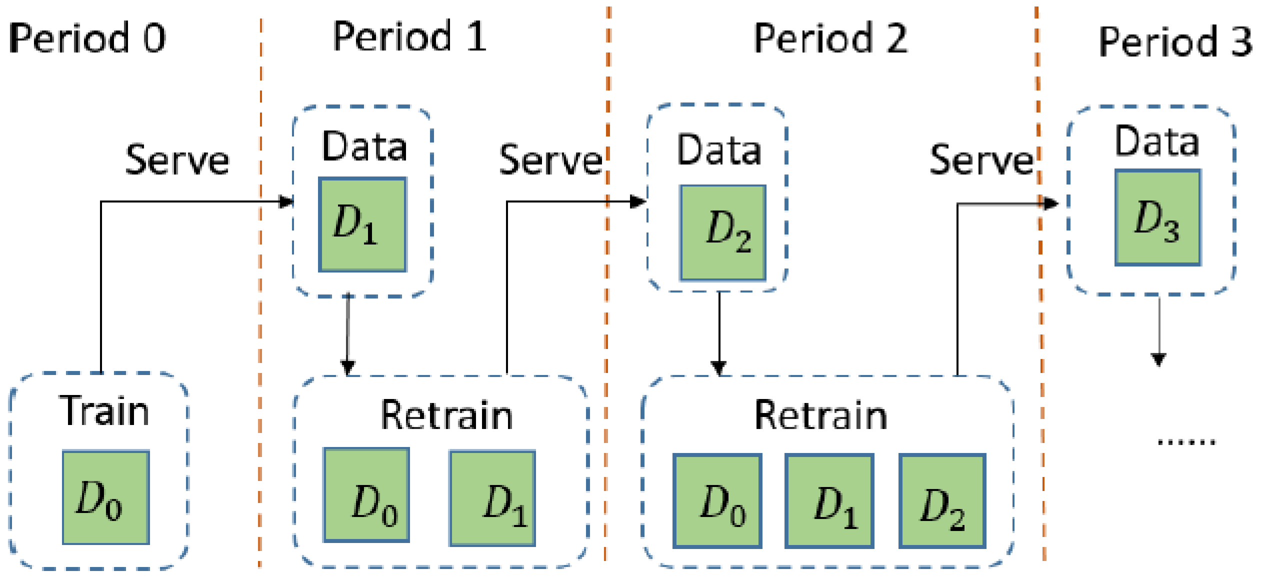


Figure 1: An illustration of periodical model retraining.

[SIGIR20]*How to Retrain Recommender System? A Sequential Meta-Learning Method*

- we propose a new retraining method with two major considerations:
- (1) building an expressive component that transfers the knowledge gained in previous training to the training on new interactions,
- (2) optimizing the transfer component towards the recommendation performance in the near future.

- the retraining of each time period is a task, which has the new interactions of the current period as the training set and the future interactions of the next period as the testing set.

$$(D_t, W_{t-1}) \xrightarrow{get} W_t \xleftarrow{test} D_{t+1},$$

METHOD

which is used for serving next period recommendations. In the t-th period retraining, W_{t-1} is set as constant input, and the retraining consists of two main steps:

1. Obtaining $\hat{W}_t$, which is expected to contain useful signal for recommendation from D_t. This step can be done by optimizing standard recommendation loss, denoted as $L_r(\hat{W}_t|D_t)$.
2. Obtaining W_t, which is the output of the transfer module:

$$W_t = f_\Theta(W_{t-1}, \hat{W}_t) \tag{3}$$

where f_Θ denotes the transfer function, Θ denotes its parameters, and W_{t-1} and $\hat{W}_t$ are its input.

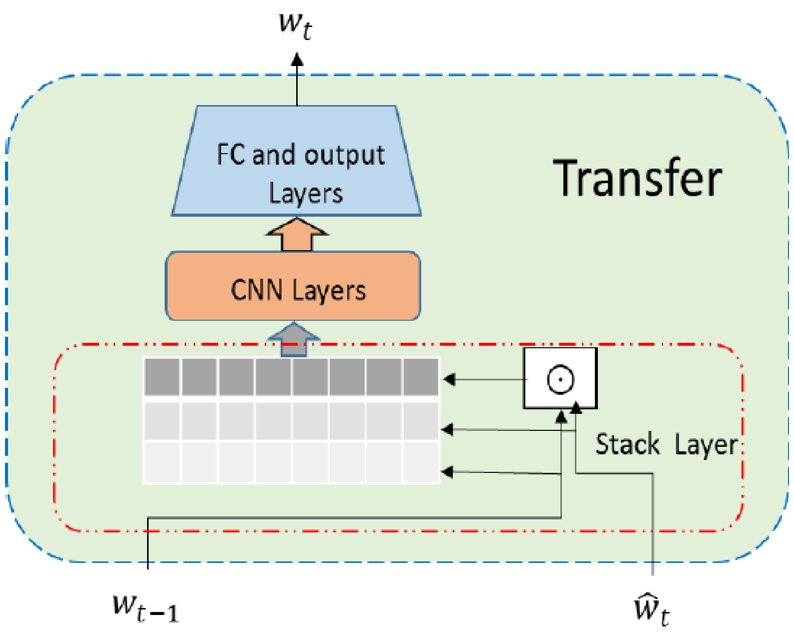


Figure 2: Model overview of our transfer-based retraining for the t-th time period. W_{t-1} represents the previous recommender, $\hat{W}_t$ is a recommender learned on new data D_t only. The transfer component is to combine the "knowledge" in W_{t-1} and $\hat{W}_t$ to obtain the new recommender W_t for serving the next period.

Transfer Design

To summarize, all trainable parameters of the transfer component are $\Theta = \{F^{(1)}, F^{(2)}, W_f, b_1, W_o, b_2\}$, where $F^{(1)} \in \mathbb{R}^{n_1 \times 3}$ and $F^{(2)} \in \mathbb{R}^{n_2 \times n_1}$ denote the filters of the first and second convolution layer,

- The CNN architecture can be found in the green box of Figure 2, which consists of a stack layer, two convolution layers, and a fully connected layer for output

Stack layer.

$$H^0 = \begin{bmatrix} w_{t-1} \\ \hat{w}_t \\ w_{dot} \end{bmatrix}, \text{where } w_{dot} = \frac{w_{t-1} \odot \hat{w}_t}{\|w_{t-1}\| + \epsilon}.$$

Convolution layers.

$$H^1_{j,m} = \mathrm{GELU}(< F_j \ , \ H^0_{:,m} >),$$

Full-connected and output layers.

$$z = \mathrm{GELU}(W_f^T \operatorname{flatten}(H^2) + b_1),$$

$$w_t = W_o^T z + b_2,$$

Sequential Training

Step 1. Learning the transfer input $\hat{W}_t$.

$$L_r(\hat{W}_t|D_t) = L_0(f_\Theta(W_{t-1}, \hat{W}_t)|D_t) + \lambda_1||\hat{W}_t||^2,$$

$$\frac{\partial L_r(\hat{W}_t|D_t)}{\partial \hat{W}_t} = \frac{\partial L_0(x|D_t)}{\partial x} \cdot \frac{\partial f_\Theta(W_{t-1}, \hat{W}_t)}{\partial \hat{W}_t} + 2\lambda_1 \hat{W}_t,$$

$$x = f_\Theta(W_{t-1}, \hat{W}_t)$$

Step 2. Learning the transfer parameter Θ.

$$L_s(\Theta|D_{t+1}) = L_0(f_\Theta(W_{t-1}, \hat{W}_t)|D_{t+1}) + \lambda_2||\Theta||^2,$$

$$\frac{\partial L_s(\Theta|D_{t+1})}{\partial \Theta} = \frac{\partial L_0(x|D_{t+1})}{\partial x} \cdot \frac{\partial f_\Theta(W_{t-1}, \hat{W}_t)}{\partial \Theta} + 2\lambda_2 \Theta,$$

$$x = f_\Theta(W_{t-1}, \hat{W}_t)$$

Algorithm 1: Sequential Training of SML

Input: Training data of T periods $\{D_t\}_{t=0}^{T}$
Output: Recommender W_T, transfer Θ

Randomly initialize W_{-1} and Θ ;
for $t = 0$ *to* T **do**
  $\hat{W}_t \leftarrow W_{t-1}$;
  while *Stop condition is not reached* **do**
    // Step 1: Learning $\hat{W}_t$
    Update $\hat{W}_t$ by optimizing $L_r(\hat{W}_t|D_t)$;
    // Step 2: Learning Θ
    if $t == T$ **then** break ;
    Update Θ by optimizing $L_s(\Theta|D_{t+1})$;
  end
  $W_t \leftarrow f_\Theta(W_{t-1}, \hat{W}_t)$;
end
return W_T, Θ

Sequential Training

Algorithm 2: Model evaluation and update

Input: Newly collected data D_{t+1}, recommender W_t to test

Output: Updated recommender W_{t+1}

1 Use D_{t+1} to test the model W_t;

2 // Model update for next period;

3 **while** *Stop condition is not reached* **do**

4   Update Θ by optimizing $L_s(\Theta|D_{t+1})$;

5   Update $\hat{W}_t$ by optimizing $L_r(\hat{W}_t|D_t)$;

6 **end**

7 Run line 4 and $W_t \leftarrow f_\Theta(W_{t-1}, \hat{W}_t)$;

8 Update $\hat{W}_{t+1}$ by optimizing $L_r(\hat{W}_{t+1}|D_{t+1})$;

9 $W_{t+1} = f_\Theta(W_t, \hat{W}_{t+1})$;

10 return W_{t+1}

Instantiation on Matrix Factorization

user-item pair (u, i), MF predicts the interaction score as:

$$\hat{y}_{ui} = p_u^T q_i,$$

- we build two separate transfer networks, one for user embedding and another for item embedding.
- Instead of feeding the embeddings of all users into the user transfer network, we operate the transfer network on the basis of each user embedding same for the item side

For each interaction $(u, i) \in D_t$, we randomly sample 1 unobserved interactions of u to form the negative data set D_t^-. Then the log loss is formulated as:

$$L_0(P, Q|D_t) = -\sum_{(u,i)\in D_t} \log(\sigma(\hat{y}_{ui})) - \sum_{(u,j)\in D_t^-} \log(1 - \sigma(\hat{y}_{uj})),$$

Table 1: **Average recommendation performance over online testing periods on Adressa and Yelp. "RI" indicates the relative improvement of SML over the corresponding baseline.**

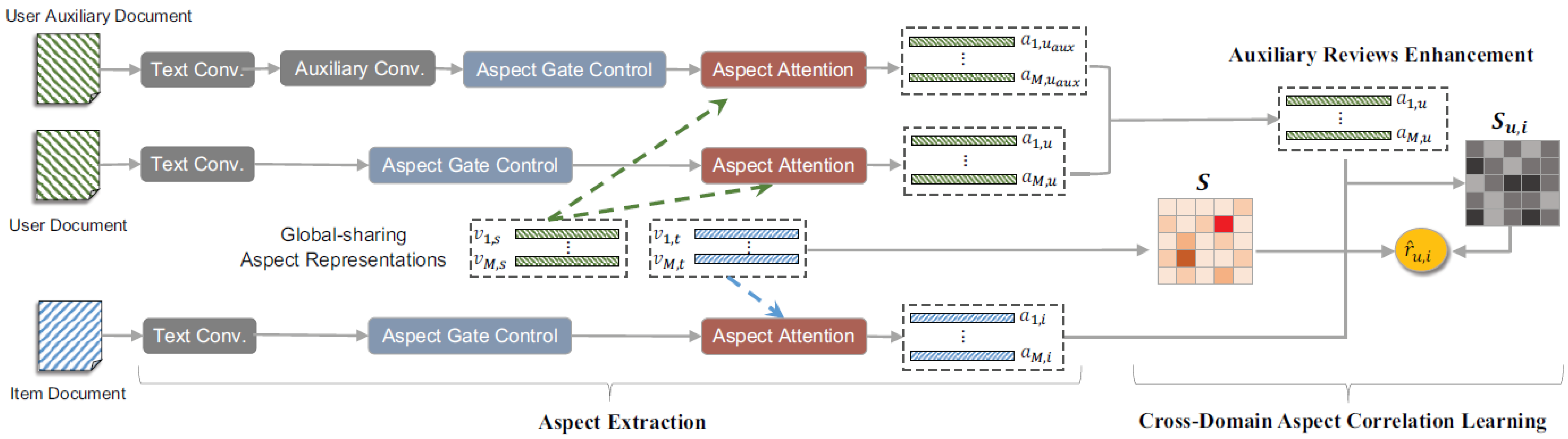
Datasets	Methods	recall@5	recall@10	recall@20	RI	NDCG@5	NDCG@10	NDCG@20	RI
Adressa	Full-retrain	0.0495	0.0915	0.1631	319.7%	0.0303	0.0437	0.0616	393.1%
	Fine-tune	0.1085	0.2235	0.3776	82.8%	0.0594	0.0962	0.1351	135.5%
	SPMF	0.1047	0.2183	0.3647	87.3%	0.0572	0.0935	0.1306	143.6%
	GRU4Rec	0.0213	0.0430	0.0860	809.0%	0.0125	0.0194	0.0302	1018.4%
	Caser	0.2658	0.3516	0.4259	6.5%	0.1817	0.2096	0.2285	2.1%
	SML	**0.2815**	**0.3794**	**0.4498**	-	**0.1838**	**0.2156**	**0.2336**	-
Yelp	Full-retrain	0.1849	0.2876	0.4139	18.0%	0.1178	0.1514	0.1829	22.7%
	Fine-tune	0.1507	0.2386	0.3534	41.7%	0.0963	0.1246	0.1535	48.5%
	SPMF	0.1664	0.2591	0.3749	30.7%	0.1072	0.1370	0.1662	35.1%
	GRU4Rec	0.1706	0.2764	0.4158	22.8%	0.1080	0.1420	0.1771	30.5%
	Caser	0.2195	0.3320	0.4565	2.8%	0.1440	0.1802	0.2117	3.12%
	SML	**0.2251**	**0.3380**	**0.4748**	-	**0.1485**	**0.1849**	**0.2194**	-

[SIGIR20]

CATN: Cross-Domain Recommendation for Cold-Start Users via Aspect Transfer Network

Cheng Zhao[1], Chenliang Li[2⋆], Rong Xiao[3], Hongbo Deng[3], Aixin Sun[4]
1. State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing, Wuhan University, Wuhan, China
2. School of Cyber Science and Engineering, Wuhan University, Wuhan, China
3. Alibaba Group, Hangzhou, China
4. School of Computer Science and Engineering, Nanyang Technological University, Singapore
{zhaocheng_whuer,cllee}@whu.edu.cn,xiaorong.xr@taobao.com,dhb167148@alibaba-inc.com,axsun@ntu.edu.sg

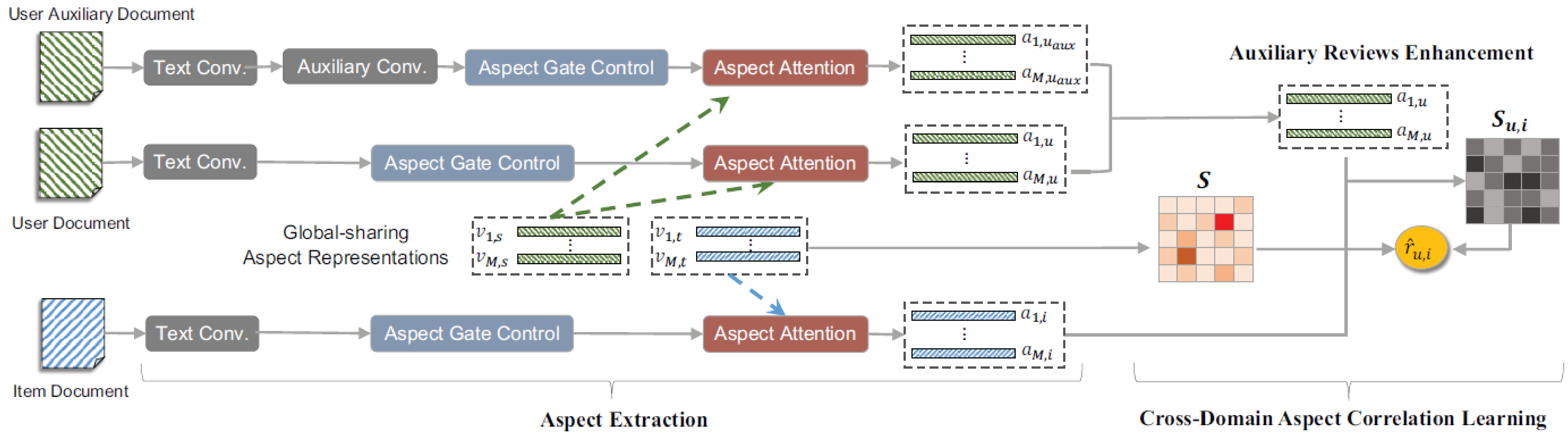
- In this paper, we propose a cross-domain recommendation framework for cold-start users via aspect transfer network, named CATN.
- In source domain, we represent a user by a user document which contains all reviews written by this user, and an item by an item document which contains all reviews it receives. The same applies in target domain. An overlapping user therefore will have two user documents, one in source domain and the other in target domain.
- To extract aspects mentioned in user and item documents, we utilize an aspect-specific gate mechanism over a convolutional layer.
- Then, global cross-domain aspect correlations are identified and weighted through attention mechanism, for preference estimation.
- To support review-based knowledge transfer, we introduce a novel cross-domain review-based preference matching procedure with two learning flows.



Text Convolution. a convolution operation with an *ReLU* activation function. $\mathbf{C}_u = [\mathbf{c}_{1,u}, \mathbf{c}_{2,u}, .., \mathbf{c}_{l,u}]$,

Aspect Gate Control. $$\mathbf{g}_{m,j,u} = (\mathbf{W}_m \mathbf{c}_{j,u} + \mathbf{b}_m) \odot \sigma(\mathbf{W}^g_m \mathbf{c}_{j,u} + \mathbf{b}^g_m)$$ $$\mathbf{G}_u = [\mathbf{G}_{1,u}, \mathbf{G}_{2,u}, .., \mathbf{G}_{M,u}],$$ $$\mathbf{G}_{m,u} = [\mathbf{g}_{m,1,u}, \mathbf{g}_{m,2,u}, .., \mathbf{g}_{m,l,u}]$$

Aspect Attention. $$\mathbf{a}_{m,u} = \sum_{j=1}^{l} \beta_{m,j,u} \mathbf{g}_{m,j,u} \qquad \beta_{m,j,u} = \frac{exp(\mathbf{g}^\top_{m,j,u} \mathbf{v}_{m,s})}{\sum_{i=1}^{l} exp(\mathbf{g}^\top_{m,i,u} \mathbf{v}_{m,s})}$$



Auxiliary Reviews Enhancement

$$c^i_{h,u_{aux}} = ReLU(\mathbf{W}^i_{aux} * \mathbf{H}_{u_{aux}}[h - \frac{s-1}{2} : h + \frac{s-1}{2}] + b^i_{aux})$$

$$\mathbf{g}_{aux} = \sigma(\mathbf{W}^1_f[(\mathbf{A}_u - \mathbf{A}_{u_{aux}}) \oplus (\mathbf{A}_u \odot \mathbf{A}_{u_{aux}})] + \mathbf{b}^1_f),$$

$$\mathbf{A}_u = tanh(\mathbf{W}^2_f[\mathbf{A}_u \oplus (\mathbf{g}_{aux} \odot \mathbf{A}_{u_{aux}})] + \mathbf{b}^2_f)$$

Cross-Domain Aspect Correlation Learning

$$\mathbf{S} = LeakyReLU(\mathbf{V}_s^\top \mathbf{W} \mathbf{V}_t)$$

$$\mathbf{S}_{u,i} = \mathbf{A}_u^\top \mathbf{W} \mathbf{A}_i$$

$$\mathbf{S}^r_{u,i} = \mathbf{S} \odot \mathbf{S}_{u,i}$$

$$\hat{r}_{u,i} = \frac{1}{M * M} \sum_{p=1}^{M} \sum_{q=1}^{M} \mathbf{S}^r_{u,i}(p,q) + b_u + b_i$$

Scenario	Scenario 1					Scenario 2					Scenario 3				
$\mathcal{D}_s \rightarrow \mathcal{D}_t$	*Book* $\rightarrow$ *Movie*					*Movie* $\rightarrow$ *Music*					*Book* $\rightarrow$ *Music*				
Method \ η	100%	50%	20%	10%	5%	100%	50%	20%	10%	5%	100%	50%	20%	10%	5%
CMF	1.167	1.169	1.179	1.179	1.181	1.139	1.140	1.158	1.167	1.173	0.939	0.942	0.962	0.967	0.970
EMCDR	1.129	1.138	1.142	1.140	1.148	1.116	1.138	1.144	1.172	1.175	0.924	0.927	0.934	0.936	0.937
CDLFM	1.126	1.130	1.135	1.138	1.144	1.115	1.133	1.145	1.169	1.171	0.918	0.925	0.930	0.931	0.951
DFM	1.141	1.143	1.149	1.150	1.156	1.136	1.158	1.162	1.166	1.175	0.923	0.929	0.933	0.941	0.952
R-DFM	1.132	1.135	1.141	1.146	1.152	1.128	1.143	1.146	1.150	1.166	0.911	0.917	0.928	0.936	0.943
ANR	1.123	1.127	1.130	1.135	1.137	1.122	1.137	1.142	1.155	1.160	0.895	0.903	0.912	0.919	0.940
CATN	**1.049**	**1.072**	**1.079**	**1.093**	**1.097**	**1.042**	**1.075**	**1.102**	**1.126**	**1.144**	**0.862**	**0.868**	**0.875**	**0.896**	**0.899**
▲%	6.59	4.88	4.51	3.70	3.52	6.55	5.45	3.50	2.09	1.38	3.69	3.88	4.06	2.50	4.06