λ**Opt**: Learn to Regularize Recommender Models in Finer Levels

Yihong Chen[†], Bei Chen[‡], Xiangnan He*, Chen Gao[†], Yong Li[†], Jian-Guang Lou[‡], Yue Wang[†]

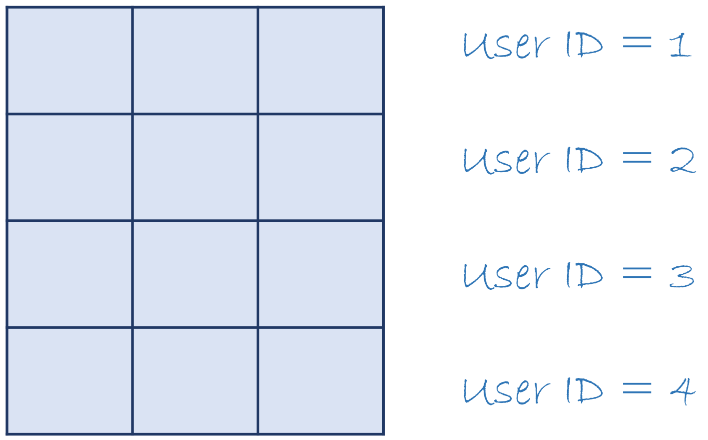
[†]Tsinghua University, [‡]Microsoft Research, *University of Science and Technology of China

Introduction

Categorical Variables in Recommender Systems

User ID
Item ID
Gender
Device Type
Buy-X-or-not
Has-Y-or-not
…
…

Categorical Variables



Generally, embedding techniques is used to handle the categorical variables.

Categorical Variables in Recommender Systems

User ID
Item ID
Gender
Device Type
Buy-X-or-not
Has-Y-or-not
...
...

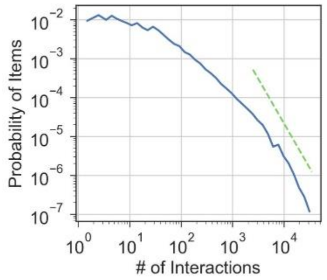
Categorical Variables

High Cardinality

Non-uniform Occurrences

Data sparsity !!!

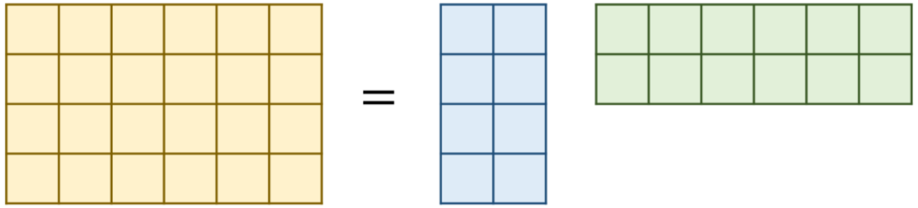
Movie IDs {1, 2, ... 4132}



Distribution of Movie ID Occurrences

Regularization Tuning Headache

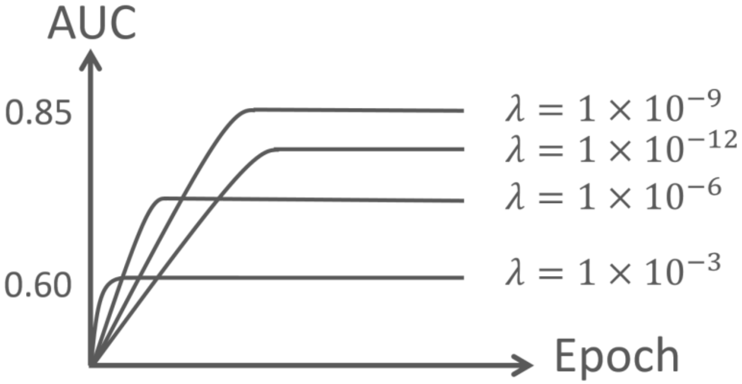
Bob



Model

$$l = \tilde{l}(\theta) + \underbrace{\lambda||\theta||^2}_{\text{Penalty}}$$

Regularized Loss



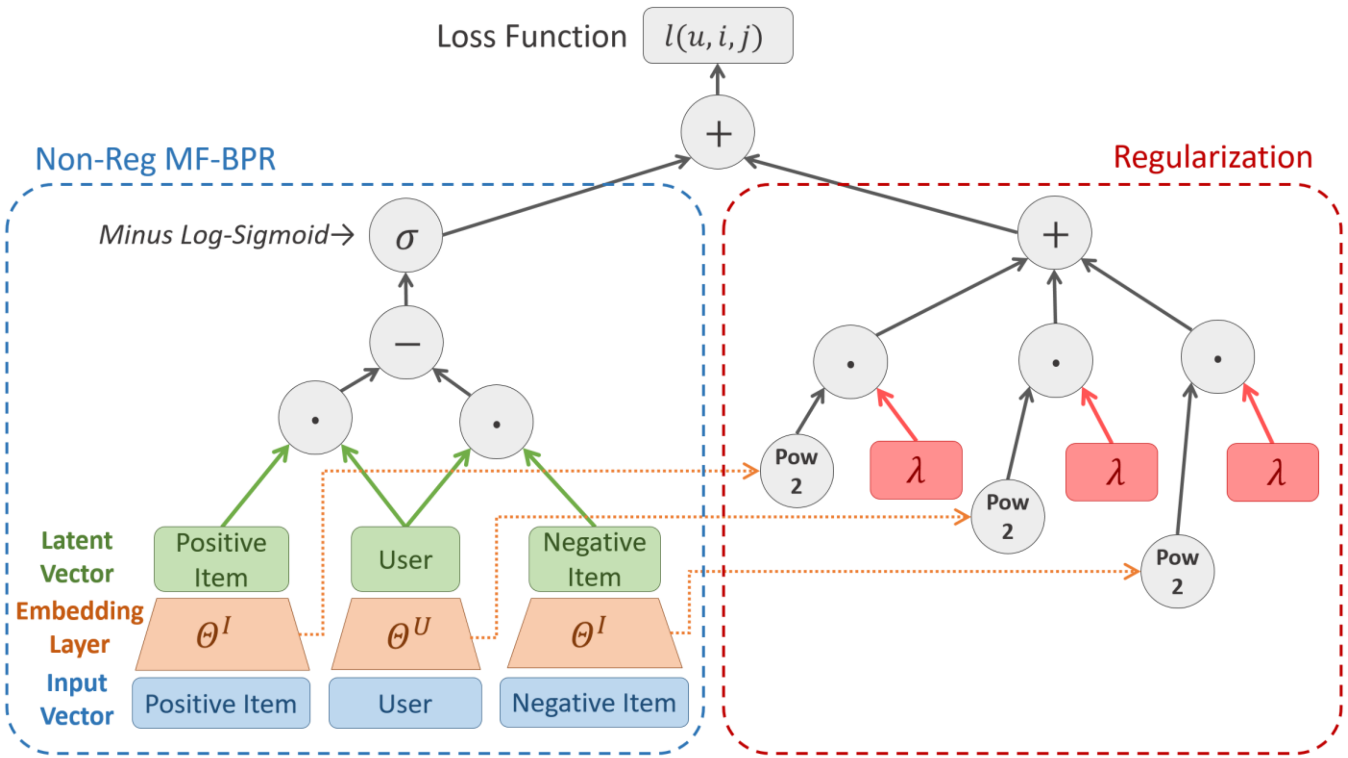
What if we can do the regularization automatically?

Related Work on Automatic Regularization for Recommender Models

- Adaptive Regularization for Rating Prediction
  - SGDA: dimension-wise & SGD based method
- Hyper-parameters Optimization
  - Grid-search, Bayesian Optimization, Neural Architecture Search $\rightarrow$ don’t specialize on recommender models’ regularization
- Regularization of Embedding
  - In NLP, training large embeddings usually suitable regularization.
  - Specific initialization methods can be viewed as some form of regularization.

Preliminaries

Matrix Factorization with Bayesian Personalized Ranking criterion



$$l_{S_T}(\Theta|\lambda) = \tilde{l}_{S_T}(\Theta) + \Omega(\Theta|\lambda)$$
$$= - \sum_{(u,i,j)\in S_T} \ln(\sigma(\hat{y}_{ui}(\Theta) - \hat{y}_{uj}(\Theta))) + \Omega(\Theta|\lambda)$$

S_T: training set,
u: user,
i: positive item,
j: negative item,
$\hat{y}_{ui}$: score function parametrized by MF for (u, i) pair
$\hat{y}_{uj}$: score function parametrized by MF for (u, j) pair

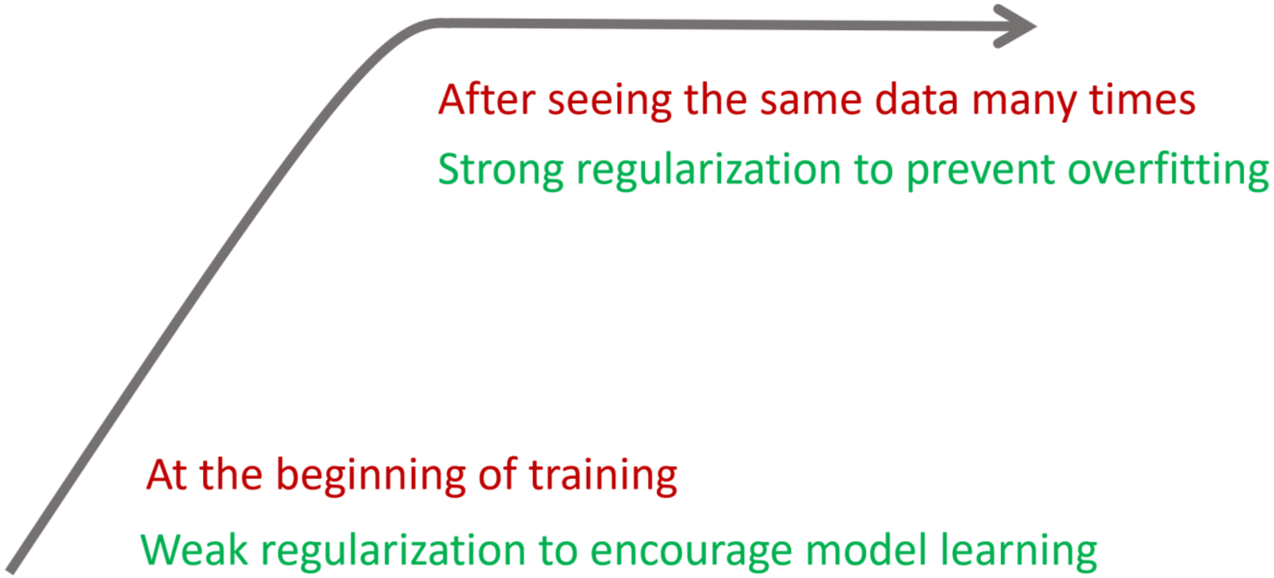
Methodology

Why hard to tune?

Hypotheses for Regularization Tuning Headache

Why hard to tune?

<u>Hypothesis 1: fixed regularization strength throughout the process</u>



Why hard to tune?

Hypothesis 2: compromise on regularization granularity

What we usually do to determine λ?

- Usually Grid Search or Babysitting $\rightarrow$ global λ

Dataset Characteristic
Diverse frequencies among users/items

Fine-grained regularization works better

- But unaffordable if we use grid-search!
- Resort to automatic methods!

Model Characteristic
Different importance of each latent dimension

How does λ**Opt** learn to regularize?

How to Train the “Brake”

Alternating Optimization to Solve the Bi-level Optimization Problem

$$\min_{\Lambda} \sum_{\{(u',i',j') \in S_V\}} l(u',i',j' \,|\, \arg\min_{\Theta} \sum_{\{(u,i,j) \in S_T\}} l(u,i,j|\Theta,\Lambda))$$

At iteration t

Train the wheel!

- Fix Λ, Optimize Θ

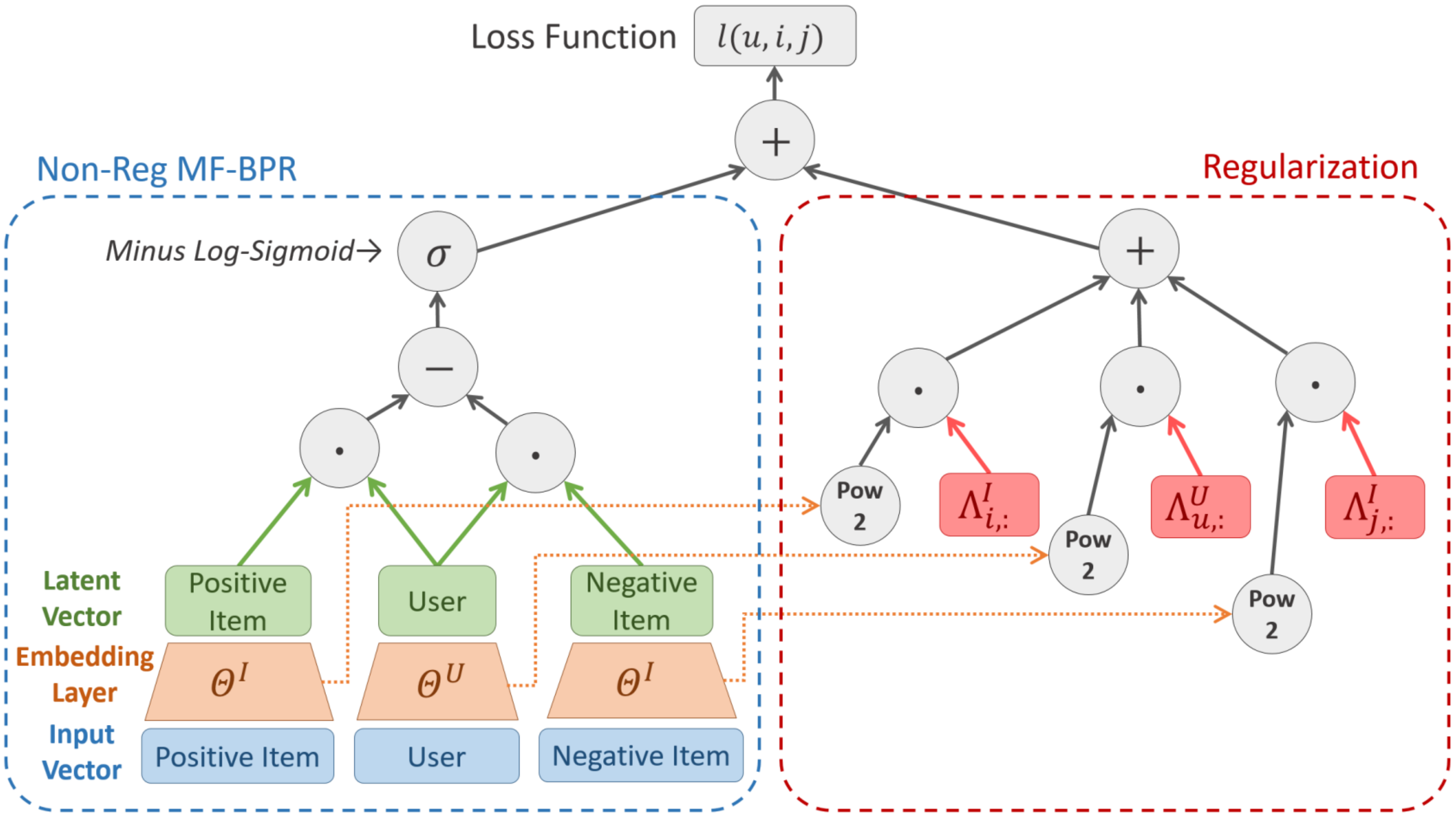
  $\rightarrow$ Conventional MF-BPR except λ is fine-grained now

Train the brake!

- Fix Θ, Optimize Λ

  $\rightarrow$ Find Λ which achieve the smallest validation loss

MF-BPR with fine-grained regularization



Fix Θ, Optimize Λ

Taking a greedy perspective, we look for Λ which can minimize the next-step validation loss

- If we keep using current Λ for next step, we would obtain $\overline{\Theta}_{t+1}$
- Given $\overline{\Theta}_{t+1}$, our aim is $\min_{\Lambda} l_{S_V}(\overline{\Theta}_{t+1})$ with the constraint of non-negative Λ

But how to obtain $\overline{\Theta}_{t+1}$ without influencing the normal Θ update?

- Simulate* the MF update!
  - Obtain the gradients by combining the non-regularized part and penalty part

$$\overline{\frac{\partial l_{S_T}}{\partial \Theta_t}} = \overline{\frac{\partial \tilde{l}_{S_T}}{\partial \Theta_t}} + \frac{\partial \Omega}{\partial \Theta_t}$$

Λ is the only variable here

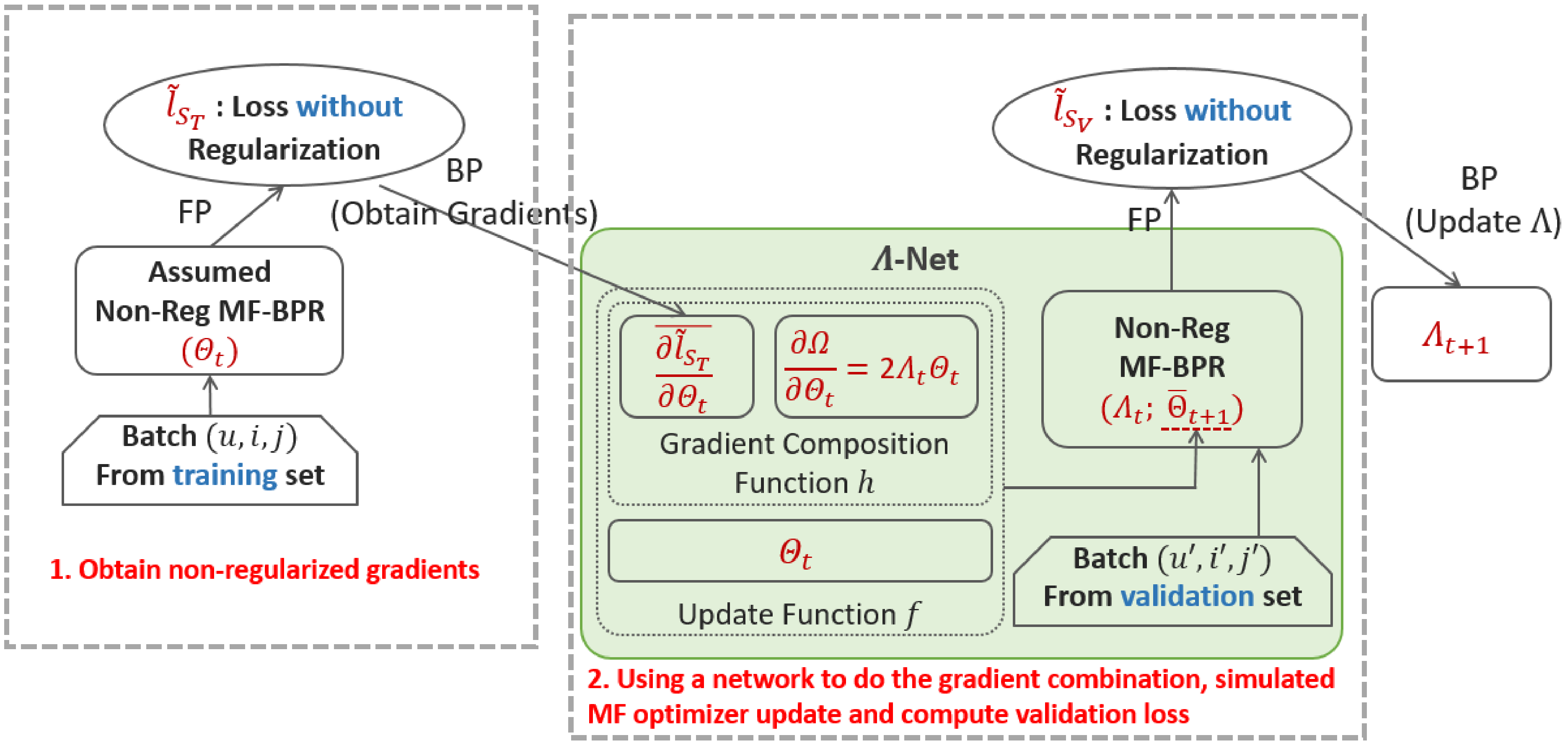
  - Simulate the operations that the MF optimizer would take

$$\overline{\Theta}_{t+1} = f(\Theta_t, \overline{\frac{\partial l_{S_T}}{\partial \Theta_t}})$$

***f* denotes the MF update function**

*: Using – over the letters to distinguish the simulated ones with normal ones

Fix Θ, Optimize Λ in Auto-Differentiation



Empirical Study

Does it really work?

Experimental settings

Datasets
- Amazon Food Review (users & items with >= 20 records)
- MovieLens 10M (users & items with >= 20 records)

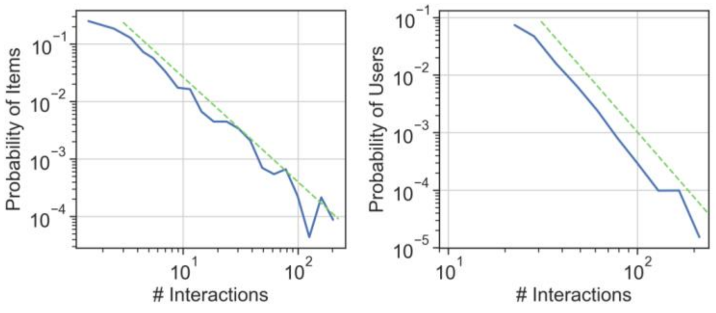
Dataset	# User	# Item	# Interaction	Density
Amazon Food	1, 238	3, 806	38, 919	0.825%
MovieLens 10M	69, 878	10, 677	10, 000, 054	1.340%

Performance measures
- **train/valid/test split: 60%, 20%, 20%**
- for each (user, item) pair in test, we make recommendations by ranking all the items that are not interacted by the user in train and valid. the truncation length K is set to 50 or 100.

Baselines
- MF-Fix: fixed global λ, choose the best after search $\lambda \in \{10, 1, 10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}, 0\}$
- SGDA (Rendle WSDM'12) dimension-wise λ + SGD optimizer for MF update
- NeuMF (He et al, WWW'17), AMF(He et al. SIGIR'18)

Variants of granularity *
- D: Dimension-wise
- DU/DI: Dimension-wise + User-wise/Dimension-wise + Item-wise
- DUI: Dimension-wise + User-wise + Item-wise

*: We use Adam optimizer for the MF update no matter what regularization granularity is



(a) Users on Amazon Food Review (b) Items on Amazon Food Review

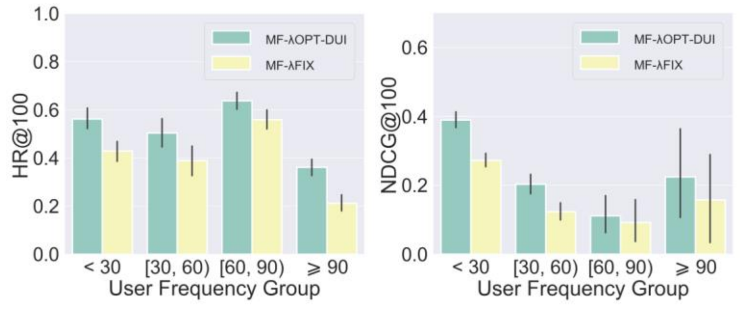
Result #1 Performance Comparison

Method		Amazon Food Review					MovieLens 10M				
		AUC	HR@50	HR@100	NDCG@50	NDCG@100	AUC	HR@50	HR@100	NDCG@50	NDCG@100
SGDA [26]		0.8130	0.1786	0.3857	0.1002	0.1413	0.9497	0.2401	0.3706	0.0715	0.0934
AMF [15]		0.8197	0.3541	0.4200	0.2646	0.2552	0.9495	0.2625	0.3847	0.0787	0.0985
NeuMF [16]		0.8103	0.3537	0.4127	0.2481	0.2218	0.9435	0.2524	0.3507	0.0760	0.0865
MF-λFix		0.8052	0.3482	0.4163	0.2251	0.2217	0.9497	0.2487	0.3779	0.0727	0.0943
MF-λOpt	-D	0.8109	0.2134	0.3910	0.1292	0.1543	0.9501	0.2365	0.3556	0.0715	0.0909
	-DU	0.8200	0.3694	0.4814	0.2049	0.2570	0.9554	0.2743	0.4109	0.0809	0.1031
	-DI	0.8501	0.2966	0.4476	0.1642	0.2039	0.9516	0.2648	0.3952	0.0804	0.1013
	-DUI	**0.8743**	**0.4470**	**0.5251**	**0.2946**	**0.2920**	**0.9575**	**0.3027**	**0.4367**	**0.0942**	**0.1158**

1. **Overall:** MF-λ**Opt**-DUI achieves the best performance, demonstrating the effect of fine-grained adaptive regularization. (approx. 10%-20% gain over baselines)
2. **Dataset:** Performance improvement on *Amazon Food Review* is larger than that on MovieLens 10M. This might due to the dataset size and density. *Amazon Food Review* has a smaller number of interactions. Complex models like NeuMF or AMF wouldn't be at their best condition. Also, smart regularization is necessary for different users/items, explaining why SGDA and MF-λ**Opt**-DUI performs worse. In our experiments, we also observe more fluctuation of training curves on *Amazon Food Review* for the adaptive λ methods.
3. **Variants of regularization granularity:** Although MF-λ**Opt**-DUI consistently performs best, MF-λ**Opt**-DU/ or MF-λ**Opt**-DU doesn't provide as much gain over the baselines, which might be due to merely addressing the regularization for partial model parameters.

Result #2: Sparseness & Activeness

Does the performance improvement come from addressing different users/items?



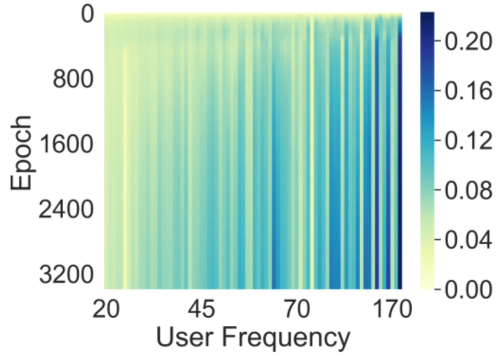
(a) HR@100 for users (b) NDCG@100 for users
(c) HR@100 for items (d) NDCG@100 for items

Group users/items according to their frequencies and check the recommendation performance of each group, using *Amazon Food Review* as an example; black line indicates variance

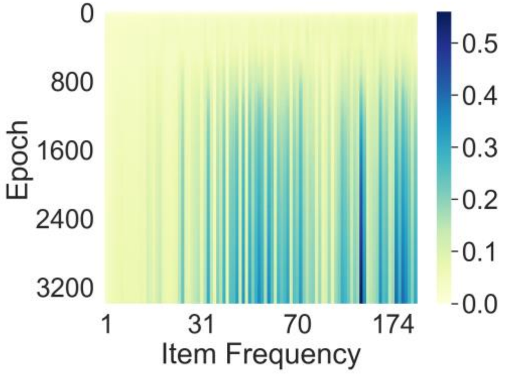
1. **<u>User with varied frequencies:</u>** For users, MF-λ**Opt**-DUI lifts HR@100 and NDCG@100. Compared to global MF-λ**Opt**-DUI , fine-grained regularization addressing users of different frequencies better.
2. **<u>Item with varied frequencies:</u>** For items, similar lift can be observed except that only slight lift for HR@100 of the <15 group and [90, 174) group.
3. **<u>Variance within the same group:</u>** Although the average lift can be observed across groups, the variance demonstrate that there are factors other than frequency which influence the recommendation performance.

Result #3: Analysis of λ-trajectory

How does MF-λ**Opt**-DUI address different users/items?



(a) For users on Amazon Food Review



For each user/item, we cache the λ from Epoch 0 to Epoch 3200 (almost converged). λs of users/items with the same frequency are averaged. The darker colors indicates larger λ.

1. **<u>λ vs. user frequency:</u>** At the same training stage, Users with higher frequencies are allocated larger λ. Active users have more data and the model learns from the data so quickly that it might get overfitting to them, making strong regularization necessary. A global λ , either small or large, would fail to satisfy both active users and sparse users.
2. **<u>It vs. item frequency:</u>** Similar as the analysis of users though not so obvious. Items with higher frequencies are allocated larger λ.
3. **<u>λ vs. training progress:</u>** As training goes on, λs gets larger gradually. Hence stronger regularization strengths are enforced at the late stage of training while the model is allowed to learn sufficiently at the beginning.

Summary

Intuition

- Fine-grained adaptive regularization

$\rightarrow$ specific λ -trajectory for each user/item

$\rightarrow$ Boost recommendation performance

Advantages

- Heterogeneous user/item in real world recommendation
- Automatically learn to regularize on-the-fly -> tuning headache
- Flexible choice in optimizers for MF models
- Theoretically generalized to other MF based models

Summary

Issues

- We observe that adaptive regularization methods are picky about the learning rates of MF update.
- Validation set size: Such validation set based methods might rely on lots of validation data. We use 20% interactions as validation set in order to make sure validation-set based methods do not overfit. This put them at advantage compared to those which don’t use validation data.
- Single-run computation cost

What’s next

- Experiments with complex matrix factorization based recommender models
- Adjust learning rate based on validation set [rather than rely on Adam]
- Study how to choose a proper validation set size

Take-away

- Fine-grained regularization (or more generally, fine-grained model capacity control) benefits recommender models
  - Due to dataset characteristics & model characteristics
  - Approximated fine-grained regularization can work well
    - Even rough approximation like greedy one-step forward

Thank you!

https://github.com/LaceyChen17/lambda-opt

Q & A