Mitigating the Popularity Bias of Graph Collaborative Filtering: *A Dimensional Collapse Perspective* (***Supplementary Material***)

Yifei Zhang[†], **Hao Zhu**[‡,§], **Yankai Chen**[†], **Zixing Song**[†], **Piotr Koniusz**[*,§,‡], **Irwin King**[†]
[†]The Chinese University of Hong Kong
[‡]Australian National University, [§]Data61♥CSIRO
{yfzhang, ykchen, zxsong, king}@cse.cuhk.edu.hk
allenhaozhu@gmail.com, piotr.koniusz@data61.csiro.au

A Appendices

A.1 Proof of Lemma 1

Proof 1 *One can set derivative of Equation (3) with respect to* $\mathbf{Z}$ *to zero and get the optimal* $\mathbf{Z}$ *as:*

$$\frac{\partial\left[\|\mathbf{Z}-\tilde{\mathbf{A}}\mathbf{E}\|_F^2+\eta\,\mathrm{tr}\left(\mathbf{Z}^T\mathbf{L}\mathbf{Z}\right)\right]}{\partial\mathbf{Z}}=0 \quad \Rightarrow \quad \mathbf{Z}-\tilde{\mathbf{A}}\mathbf{E}+\eta\mathbf{L}\mathbf{Z}=0. \tag{15}$$

Note that $\det(\mathbf{I}+\eta\mathbf{L})>0$*, thus matrix* $\{\mathbf{I}+\eta\mathbf{L}\}^{-1}$ *exists. Then the corresponding closed-form solution can be written as:*

$$\mathbf{Z}=\left((1+\eta)\mathbf{I}-\eta\tilde{\mathbf{A}}\right)^{-1}\tilde{\mathbf{A}}\mathbf{E} \tag{16}$$

Since $\frac{\eta}{1+\eta}<1$ *for* $\forall\eta>0$*, and matrix* $\tilde{A}$ *has absolute eigenvalues bounded by 1, thus, all its positive powers have bounded operator norm, then the inverse matrix can be decomposed as follows with* $k\to\infty$ *:*

$$\begin{aligned}\mathbf{Z}&=\frac{1}{1+\eta}\left(\mathbf{I}-\frac{\eta}{1+\eta}\tilde{\mathbf{A}}\right)^{-1}\tilde{\mathbf{A}}\mathbf{E}\\&=\frac{1}{1+\eta}\left(\mathbf{I}+\frac{\eta}{1+\eta}\tilde{\mathbf{A}}^1+\frac{\eta^2}{(1+\eta)^2}\tilde{\mathbf{A}}^2+\cdots+\frac{\eta}{(1+\eta)^K}\tilde{\mathbf{A}}^K+\cdots\right)\tilde{\mathbf{A}}\mathbf{E}\\\mathbf{Z}&=\frac{1}{1+\eta}\tilde{\mathbf{A}}\mathbf{E}+\frac{\eta}{(1+\eta)^2}\tilde{\mathbf{A}}^2\mathbf{E}+\cdots+\frac{\eta^{K-1}}{(1+\eta)^K}\tilde{\mathbf{A}}^K\mathbf{E}+\cdots\end{aligned} \tag{17}$$

Note that $\frac{1}{1+\eta}+\frac{\eta}{(1+\eta)^2}+\cdots+\frac{\eta^{K-1}}{(1+\eta)^K}+\cdots=1$ *and we can change the coefficient* $\eta\in(0,\infty)$ *to fit fusion weights* $\alpha_1,\alpha_2,\cdots,\alpha_K$*. When the layer* K *is large enough, the propagation mechanism of LightGCN in Equation (3) approximately corresponds to the objective Equation (4).*

[*]The corresponding author. Code: https://github.com/yifeiacc/LogDet4Rec/

A.2 Proof of Lemma 2

Proof 2 *First, let us take the gradient of* $\operatorname{tr}\left(\mathbf{Z}^\top \mathbf{L}\mathbf{Z}\right)$ *with respect to the input matrix* $\mathbf{E}$ *and denote* $\mathbf{Z} = \hat{\mathbf{A}}\mathbf{E}$ *where* $\hat{\mathbf{A}} = \sum_{k=1}^{K} \alpha_k \tilde{\mathbf{A}}^k$.

$$\begin{aligned}\frac{\partial \mathcal{L}_{smooth}}{\partial \mathbf{E}} &= \frac{\partial \operatorname{tr}\left(\mathbf{Z}^\top \mathbf{L}\mathbf{Z}\right)}{\partial \mathbf{E}} \\ &= \frac{\partial \operatorname{tr}\left((\hat{\mathbf{A}}\mathbf{E})^\top \mathbf{L}(\hat{\mathbf{A}}\mathbf{E})\right)}{\partial \mathbf{E}} \\ &= 2\hat{\mathbf{A}}^\top \mathbf{L}\hat{\mathbf{A}}\mathbf{E} \\ &= 2\mathbf{Q}\mathbf{E}.\end{aligned} \tag{18}$$

Treat the weight matrix as a function of the training step t, i.e., $\mathbf{E} = \mathbf{E}(t)$, *then we can derive the gradient of* $\mathbf{E}(t)$ *with respect to* t *by* $\frac{\mathrm{d}\mathbf{E}(t)}{\mathrm{d}t} = 2\mathbf{Q}\mathbf{E}$. *As both* $\mathbf{Q}$ *are fixed, we can solve the equation analytically,*

$$\mathbf{E}(t) = \exp(2\mathbf{Q}t) \cdot \mathbf{E}(0). \tag{19}$$

As we have the non-ascending eigenvalues of $\mathbf{Q}$ *as* $\lambda_1^{\mathbf{Q}} \geq \lambda_2^{\mathbf{Q}} \geq \cdots \geq \lambda_D^{\mathbf{Q}}$, *we can define an auxiliary function* $f\left(t; \lambda_i^{\mathbf{Q}}, \lambda_j^{\mathbf{Q}}\right) = \exp\left(\lambda_i^{\mathbf{Q}} t\right) / \exp\left(\lambda_j^{\mathbf{Q}} t\right) = e^{(\lambda_i^{\mathbf{Q}} - \lambda_j^{\mathbf{Q}})t}$. *It is obvious that* $f\left(t; \lambda_i^{\mathbf{Q}}, \lambda_j^{\mathbf{Q}}\right)$ *is monotonically decreasing for all* $i > j$. *As* $\mathbf{E}(t)$ *is a transformation of its initial state* $\mathbf{E}(0)$ *up to* $\exp(\mathbf{Q}t)$, *we can conclude that:*

$$\frac{\sigma_i^{\mathbf{E}}(t)}{\sigma_j^{\mathbf{E}}(t)} \leq \frac{\sigma_i^{\mathbf{E}}(t')}{\sigma_j^{\mathbf{E}}(t')}, \quad \forall t < t' \ \text{ and } \ i > j.$$

Let the spectrum be following the descending order. Then we have $\lim_{t\to\infty} f\left(t; \lambda_i^{\mathbf{Q}}, \lambda_j^{\mathbf{Q}}\right) = 0, \forall i > j$ *if* $\lambda_i^{\mathbf{Q}} \neq \lambda_j^{\mathbf{Q}}$.

Notice the above expression analyses the decay of spectrum for matrix $\exp(2\mathbf{Q}t)$. *Thus, assume* $\mathbf{E}(0)$ *is a full-rank matrix. Then*

$$\operatorname{rank}(\exp(2\mathbf{Q}t) \cdot \mathbf{E}(0)) \leq \min\left[\operatorname{rank}(\exp(2\mathbf{Q}t), \operatorname{rank}(\mathbf{E}(0))\right]$$

due to the well-known inequality stating that $\operatorname{rank}(\mathbf{X}\mathbf{Y}) \leq \min\left(\operatorname{rank}(\mathbf{X}), \operatorname{rank}(\mathbf{Y})\right)$.

A.3 Proof of Corollary 2

Imagine that $\mathbf{\Sigma}_{\mathcal{U}} = \operatorname{diag}([1, 0.1])$. Let $\Delta\mathbf{\Sigma}_{\mathcal{U}} = \operatorname{diag}([0, -0.1])$ then $\log\det(\mathbf{\Sigma}_{\mathcal{U}} + \mathbf{\Sigma}_{\mathcal{U}}) = \log\lambda_1 + \log\lambda_2 = \log 1 + \log 0 = -\infty$ so $\mathcal{L}_{logdet} = \infty$. In contrast, for $\mathcal{L}_{soft}$ we have $(\lambda_1 - 1)^2 + (\lambda_2 - 1)^2 = 0.81$. If 10 user feature vectors $f(u) = diag([1, 0.31])$ are in relation with 10 item feature vectors $f(i) = \operatorname{diag}([1, 0.0])$, it is easy to see that $10 \cdot 0.31^2 \approx 1$ which means the alignment loss is better off with the dimensional collapse for $\mathcal{L}_{soft}$ as $1 > 0.81$. For the LogDet penalty however we have $1 \ll \infty$.

A.4 Proof of Lemma 3

We have the following:

Proof 3

$$D_{\phi_{ld}}(\mathbf{X}, \mathbf{I}) = \operatorname{tr}(\mathbf{X}) - \log\det(\mathbf{X}) - d = \sum_{i=1}^{d} \left(\lambda_i - \log\lambda_i - 1\right). \tag{20}$$

Now, $x - \log x \geq 1$ *with equality at* $x = 1$. *Also,* $x - \log x \geq \log x + 1 - \log 4$ *with equality at* $x = 2$. *Letting* $\lambda_1 \geq \lambda_2 \cdots \geq \lambda_d > 0$, *we have:*

$$\begin{aligned} & D_{\phi_{ld}}(\mathbf{X}, \mathbf{I}) \geq (\log\lambda_1 + 1 - \log 4) - (\log\lambda_d + 1) \\ \Longrightarrow \quad & \operatorname{Cond}(\mathbf{X}) \leq 4\exp D_{\phi_{ld}}(\mathbf{X}, \mathbf{I}) \end{aligned} \tag{21}$$

Thus, LogDet yields an upper bound on the condition number.

A.5 Detailed Settings

For the general settings, we create the user and item embeddings with the Xavier initialization of dimension 64; we use Adam to optimize all the models with the learning rate 0.001; the l_2 regularization coefficient 10^{-4} and the batch size 2048 are used, which are common in many papers [15, 43, 42]. In SimGCL and SGL, we empirically set the temperature $\tau = 0.2$ as this value is often reported the best choice in papers on CL [43, 40]. An exception is that we let $\tau = 0.15$ for XSimGCL on Yelp2018, which brings a slightly better performance. Note that although the paper of SGL [43] uses Yelp2018 and Alibaba-iFashion as well, we cannot reproduce their results on Alibaba-iFashion with their given hyperparameters under the same experimental setting. So we re-search the hyperparameters of SGL and choose to present our results on this dataset in Table 3.

A.6 Dataset Statistics

Table 6: Dataset statistics.

Dataset	#User	#Item	#Feedback	Density
Yelp2018	31,668	38,048	1,561,406	0.13%
Amazon-Kindle	138,333	98,572	1,909,965	0.014%
Alibaba-iFashion	300,000	81,614	1,607,813	0.007%

A.7 Effective Rank

Definition 2 (Effective Rank.) *Consider matrix $\boldsymbol{X} \in \mathbb{R}^{m \times n}$ whose singular value decomposition is given by $\boldsymbol{X} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T$, where $\boldsymbol{\Sigma}$ is a diagonal matrix with singular values $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_Q \geq 0$ with $Q = \min\{m, n\}$. The distribution of singular values is defined as the ℓ_1-norm normalized form $p_i = \sigma_i / \sum_{k=1}^{Q} |\sigma_k|$. The effective rank of the matrix $\boldsymbol{X}$, denoted as erank $(\boldsymbol{X})$, is defined as* $\operatorname{erank}(\boldsymbol{X}) = \exp\left(H\left(p_1, p_2, \cdots, p_Q\right)\right)$*, where $H\left(p_1, p_2, \cdots, p_Q\right)$ is the Shannon entropy given by* $H\left(p_1, p_2, \cdots, p_Q\right) = -\sum_{k=1}^{Q} p_k \log p_k$.

A.8 Comparison of Runtimes with the Riemannian Metric

The main advantage of $D_{\phi_{ld}}$ over the Riemannian metric D_R (such as AIRM [10] and LERM [10]) is its computational speed. To compute $D_{\phi_{ld}}$, only determinants need to be computed, which can be efficiently achieved with Cholesky factorization (for $\boldsymbol{\Sigma}_{\mathcal{U}}\boldsymbol{\Sigma}_{\mathcal{I}}$) at a cost of $\frac{1}{3}d^3$ flops [12]. On the other hand, computing the Riemannian metric requires generalized eigenvalues, which takes around $4d^3$ flops for positive definite matrices. Therefore, in general, $D_{\phi_{ld}}$ is much faster to compute (see Table 7b). This speed advantage becomes even more pronounced when computing gradients. Moreover, backpropagation through the matrix determinant is generally stable whereas generalized eigenvalue decomposition suffers undetermined gradients if two eigenvalues are non-simple (equal values). As shown in Table 7a, computing $\partial D_{\phi_{ld}}$ can be over 100 times faster than ∂D_{ϕ_R}. This difference can be crucial when using gradient-based algorithms, such as neural networks, that rely on the computation of similarity measure gradients.

Table 7: Runtime computed over 1000 trials (millisecs/trial).

(a) Average times to compute gradients.

d	$\partial_{\mathbf{X}} D_R(\mathbf{X}, \mathbf{I})$	$\partial_{\mathbf{X}} D_{\phi_{ld}}(\mathbf{X}, \mathbf{I})$
5	0.79815$\pm$0.0934	0.036$\pm$0.009
10	2.38341$\pm$0.2094	0.058$\pm$0.021
20	7.49365$\pm$0.5954	0.110$\pm$0.013
40	24.8942$\pm$1.1264	0.270$\pm$0.047
80	99.4825$\pm$5.1813	0.921$\pm$0.028
200	698.813$\pm$39.602	8.767$\pm$2.137
500	6377.22$\pm$379.11	94.83$\pm$1.195
1000	40443.0$\pm$2827.2	622.2$\pm$37.70

(b) Average times to compute function values.

d	$D_R(\mathbf{X}, \mathbf{I})$	$D_{\phi_{ld}}(\mathbf{X}, \mathbf{I})$
5	0.025 $\pm$ 0.012	0.030$\pm$0.007
10	0.036 $\pm$ 0.005	0.040$\pm$0.009
20	0.085 $\pm$ 0.006	0.061$\pm$0.009
40	0.270 $\pm$ 0.332	0.123$\pm$0.012
80	1.234 $\pm$ 0.055	0.393$\pm$0.050
200	8.198 $\pm$ 0.129	2.223$\pm$0.169
500	77.311 $\pm$ 0.568	22.18$\pm$1.223
1000	492.743 $\pm$ 15.51	119.7$\pm$1.416

A.9 Performance Comparison w.r.t. to Different Distance Types

Table 8: Main comparison on different distances. ♣ denotes the Matrix Norm and ♠ denotes the Bregman Matrix Divergence. KL Matrix Div. is Bergman Div. associated with $\phi(\mathbf{X}) = \sum_i \lambda_i \log \lambda_i$.

Geometries	$D(\mathbf{\Sigma}_\mathcal{X}, \mathbf{I})$	**Yelp2018**		**iFashion**	
		Recall@20	**NDCG@20**	**Recall@20**	**NDCG@20**
Euclidean Norm ♣	$\|\mathbf{\Sigma}_\mathcal{X} - \mathbf{I}\|_2$	0.0563	0.0459	0.0890	0.0404
Nuclear Norm ♣	$\|\mathbf{\Sigma}_\mathcal{X} - \mathbf{I}\|_*$	0.0632	0.0516	0.0998	0.0458
Frobenius Norm ♣♠	$\|\mathbf{\Sigma}_\mathcal{X} - \mathbf{I}\|_F$	0.0709	0.0592	0.1112	0.0580
KL Matrix Div. ♠	$\mathrm{tr}\left(\mathbf{\Sigma}_\mathcal{X} \log \mathbf{\Sigma}_\mathcal{X}\right)$	0.0724	0.0602	0.1110	0.0597
Logdet Div. ♠	$-\log\det\left(\mathbf{\Sigma}_\mathcal{X}\right)$	**0.0732**	**0.0618**	**0.1270**	**0.0617**

In this section, we adopt other types of distances as in Equation (11) and compare them with proposed method. There are two different categories (1) the matrix norm, *i.e.*, Euclidean Norm, Nuclear Norm, and Forbenius Norm, and (2) the Bregman divergence, *i.e.*, von Neumann divergence (also known as Matrix Kullback–Leibler divergence) and Logdet divergence. The Forbenius norm can belong to both matrix norm and Bregman Matrix divergence. We show their formulas and experimental result in Table 8. We notice that the proposed method (Logdet Div.) achieves the best results.

Table 9: Result for Recall@5, Recall@10, and Recall@20.

Dataset	Model	Recall@5	Recall@10	Recall@20
Yelp2018	GCF$_loget$	0.0275	0.0445	0.0732
	DirectAU	0.0255	0.0426	0.0720
	LightGCN	0.0211	0.0336	0.0590
iFashion	GCF$_loget$	0.0301	0.0483	0.0617
	DirectAU	0.0292	0.0493	0.0601
	LightGCN	0.0284	0.0391	0.0484

A.10 Broader impact and limitations

Our method enjoys impact and limitations similar to those in graph collaborative filtering. Typical GCF models cannot guarantee they can utilize the feature space efficiently. The mode collapse can lead to frequent item bias resulting in a model which is biased in its recommendations. Thus, by improving recommendation of rare items we also offer a generic approach to limiting the recommendation bias which is important in many domains of life. Our method requires no special resources and just a fraction of additional computations beyond what GCF uses. Of course, our model is limited by the GCF model itself and it is applicable only to pipelines that suffer the mode collapse.