Low-rank Matrix Approximation with Stability

2019. 02. 19.

Joonyoung Yi
joonyoung.yi@kaist.ac.kr

KAIST

CONTENTS

CONTENTS

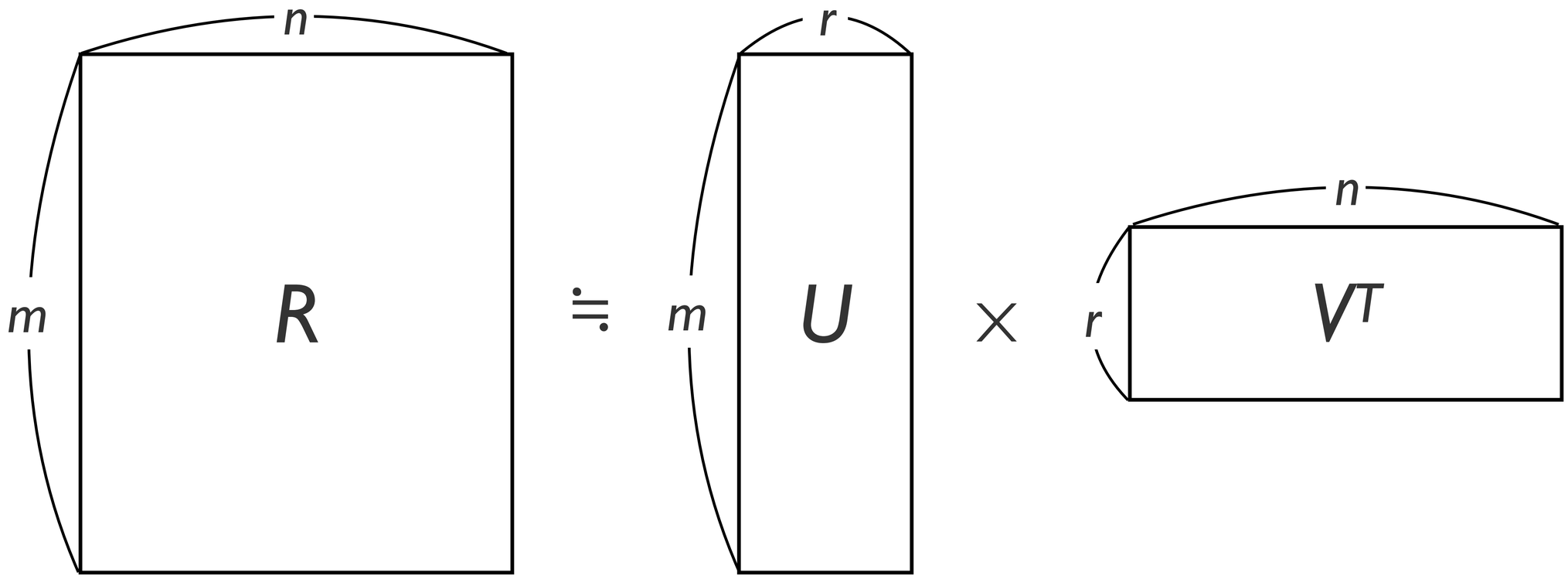
KAIST

MLILAB
Machine Learning & Intelligence

Low-rank Matrix Factorization (LRMF)

- Let $R \in \mathbb{R}^{m \times n}, U \in \mathbb{R}^{m \times r}, V \in \mathbb{R}^{n \times r}$ are matrices.
- **Low-rank Matrix Factorization** Problem:
  We know only 1% of entries in R, how to complete low-rank matrix R?
- Optimization form:

$$\hat{R} = \arg\min_{X} Loss(R, X),\ rank(X) = r.$$



Low-rank Matrix Approximation

- In vanilla Low-rank Matrix Factorization Problem,
  a solution find one pair of *U* and *V* **(single model)**.
- To enhance performance of single model,
  we can use **ensemble methods**.
- We can say some algorithm is low-rank matrix approximation
  when it finds multiple low-rank matrices to complete final matrix.

- For example,
  - [ICML'13] LLoRMA: Local Low-rank Matrix Approximation
  - [NIPS'17] MRMA: Mixture Rank Matrix Approximation
  - [AAAI'17] GLOMA: Global Information in Local Matrix Approximation
  - **[ICML'16] SMA: Low-rank Matrix Approximation with Stability**

Motivation of This Paper

- The Matrix *R* we have to complete is very sparse.
  - We only know less than 1% of total entries.

- Thus, single out-of-distribution entry may **significantly change** the entire matrix.
- For example,

$$\begin{bmatrix} 1 & 1 \\ 2 & ? \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \end{bmatrix} \times \begin{bmatrix} 1 & 1 \end{bmatrix}$$

2

$$\begin{bmatrix} 1 & 4 \\ 2 & ? \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \end{bmatrix} \times \begin{bmatrix} 1 & 4 \end{bmatrix}$$

8

- Nowadays, due to the data poisoning attack, stability is becoming more important.
  - Data Poisoning Attack on Collaborative Filtering [NIPS'16, '18]: Promote and demote specific items by creating many accounts that.

EXISTING WORKS

- [NIPS'07] Probabilistic Matrix Factorization

$$\hat{R} = \arg\max_{U,V} Loss(R, UV^T) + \lambda(\|U\|_F^2 + \|V\|_F^2)$$

  - To handle noise by adding regularization terms
  - Generalization error is too high.
- Koren handle stability issue by cross validation.
  - Cross-validation technique have drawback that the amount of data available for model learning is reduced.
- [ICML'13] LLoRMA suggested ensemble technique.
  - Ensemble methods are computationally expensive due to the training sub models.

- →This paper will suggests stable ensemble method which is not computationally expensive.

CONTENTS

KAIST

MLILAB
Machine Learning & Intelligence

Overview of This Paper

- To suggest stable matrix approximation method: SMA
  - First introduce the stability notion in LRMA.
  - Develop theoretical guidelines for deriving LRMA solutions with high stability.
  - Formulate a new optimization problem for achieving stable LRMA
  - Develop a stochastic gradient descent method to solve the new optimization problem.
  - Show effectiveness in real datasets (Movielens, Netflix)

- The key contributions
  - (1) First introduces the stability concept in LRMA.
  - (2) Proposes a stable LRMA algorithm with high stability and high generalization performance.
  - (3) Makes significant improvement in prediction accuracy.

What is Algorithmic Stability?

- **Replacing one element** in the training set
  does not result in significant change to the algorithm's output
  in a stable learning algorithm.
  - Bousquet & Elisseeff (2001), Lan et al., (2008) London et al., (2013) demonstrated.
  - Var[Training error] is small.
  - Training error ≈ Test error

- This paper introduce this concepts to LRMA algorithm.

Stability w.r.t. Matrix Approximation

- Root Mean Square Error (RMSE) is a common evaluation metric for recommendation tasks.

$$\mathcal{D}(\hat{R}) = \sqrt{\frac{1}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n}(R_{i,j} - \hat{R}_{i,j})^2}\ ,\ \mathcal{D}_{\Omega}(\hat{R}) = \sqrt{\frac{1}{|\Omega|}\sum_{(i,j)\in\Omega}(R_{i,j} - \hat{R}_{i,j})^2}$$

Definition 1

- For any $R \in \mathbb{R}^{m\times n}$, Choose a subset of entries Ω from *R* uniformly.
- For a given $\epsilon > 0$, $\mathcal{D}_{\Omega}(\hat{R})$ is δ-stable if

$$\Pr\left[|\mathcal{D}(\hat{R}) - \mathcal{D}_{\Omega}(\hat{R})| \leq \epsilon\right] \geq 1 - \delta$$

- The generalization error is bounded with high probability.
- Minimizing the training error will induce minimizing test error.

Stability w.r.t. Matrix Approximation

Definition 1

- For any $R \in \mathbb{R}^{m \times n}$, Choose a subset of entries Ω from *R* uniformly.
- For a given $\epsilon > 0$, $\mathcal{D}_\Omega(\hat{R})$ is δ-stable if

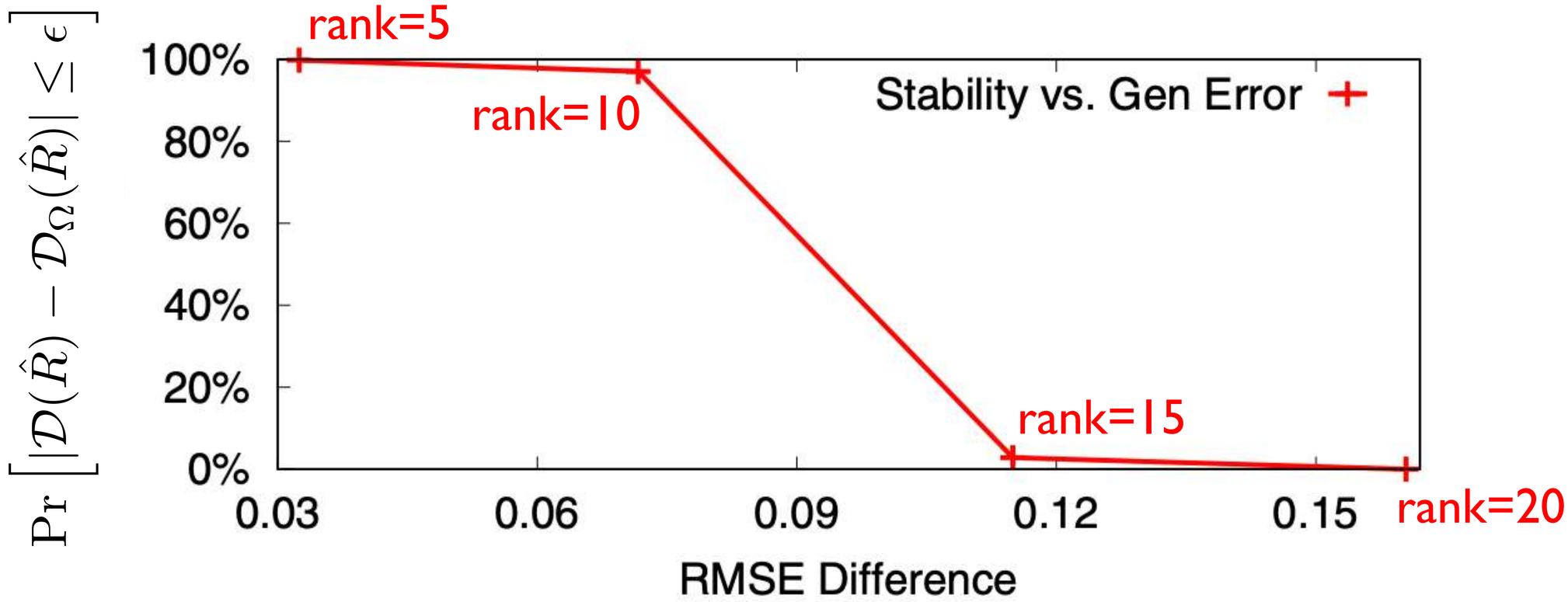
$$\Pr\left[|\mathcal{D}(\hat{R}) - \mathcal{D}_\Omega(\hat{R})| \leq \epsilon\right] \geq 1 - \delta$$

- For example,
  - Any two subsets of entries Ω_1, Ω_2 from *R*.
  - Approximating *R* by Ω_1 is δ_1-stable.
  - Approximating *R* by Ω_2 is δ_2-stable.
  - $\mathcal{D}_{\Omega_1}(\hat{R})$ is more stable than $\mathcal{D}_{\Omega_2}(\hat{R})$ if $\delta_1 < \delta_2$.
  - Minimizing $\mathcal{D}_{\Omega_1}(\hat{R})$ leads higher generalization performance than $\mathcal{D}_{\Omega_2}(\hat{R})$.

- Smaller δ means more stable.

STABILITY VS. GENERALIZATION ERROR

- Experiments on Movielens 1M dataset using RSVD (Patrek, 2007).



- In the existing LRMA method,
  - The test error becomes lower as the rank increases.
  - However, the **stability becomes lower** and the **generalization error increases** as the rank increases.
  - Can't provide stable recommendation even the rank is as low as 20.

Key Finding of This Paper

- $\Omega' \leftarrow$ remove a subset of easily predictable entries from Ω.
- Minimizing both $\mathcal{D}_\Omega$ and $\mathcal{D}_{\Omega'}$ together will make **more stable solution** than the solution of minimizing $\mathcal{D}_\Omega$ **only**.
- Formally,

Theorem 1

- Let (a set of easily predictable entries) $\omega \subset \Omega(|\Omega| > 2)$ which satisfies that

$$\forall (i,j) \in \omega, |R_{i,j} - \hat{R}_{i,j}| \leq \mathcal{D}_\Omega(\hat{R})$$

- Let (a set of entries hard to predict) $\Omega' = \Omega - \omega$.
- For any $\epsilon > 0$ *and* $1 > \lambda_0, \lambda_1 > 0$ *(*$\lambda_0 + \lambda_1 = 1$*),*
- $\lambda_0 \mathcal{D}_\Omega(\hat{R}) + \lambda_1 \mathcal{D}_{\Omega'}(\hat{R})$ *and* $\mathcal{D}_\Omega(\hat{R})$ are δ_1-stable and δ_2-stable respectively.
- Then, $\delta_1 \leq \delta_2$

Key Finding of This Paper

Hoeffding's Lemma

- Let $X \in \mathbb{R}$ be random variable.
- E[X] = 0, and $\Pr[X \in [a, b]] = 1$.
- Then, for any $s \in \mathbb{R}$, $\mathrm{E}\left[e^{sX}\right] \leq \exp\left(\frac{1}{8}s^2(b-a)^2\right)$

- (Proof Sketch of **Theorem 1**)
- Let $\mathcal{D}(\hat{R}) - \mathcal{D}_\Omega(\hat{R}) \in [-a, a]$
- By Markov's inequality $\Pr[\mathcal{D}(\hat{R}) - \mathcal{D}_\Omega(\hat{R}) \geq \epsilon] \leq \frac{E(e^{t(\mathcal{D}(\hat{R}) - \mathcal{D}_\Omega(\hat{R}))})}{e^{t\epsilon}}$
- Based on Hoeffding's Lemma,

  $\Pr[\mathcal{D}(\hat{R}) - \mathcal{D}_\Omega(\hat{R}) \geq \epsilon] \leq \frac{\exp\left(\frac{1}{2}t^2a^2\right)}{\exp\left(t\epsilon\right)}$

  $\Pr[-\mathcal{D}(\hat{R}) + \mathcal{D}_\Omega(\hat{R}) \geq \epsilon] \leq \frac{\exp\left(\frac{1}{2}t^2a^2\right)}{\exp\left(t\epsilon\right)}$.
- Similar steps on $\lambda_0\mathcal{D}_\Omega(\hat{R}) + \lambda_1\mathcal{D}_{\Omega'}(\hat{R})$
- Via simple calculations, we can prove **Theorem 1**.

Condition Relaxation

- From **Theorem 1**, we know that minimizing $\lambda_0 \mathcal{D}_\Omega(\hat{R}) + \lambda_1 \mathcal{D}_{\Omega'}(\hat{R})$ is better than minimizing $\mathcal{D}_\Omega(\hat{R})$ **only**.
- However, the condition of $\forall(i,j) \in \omega, |R_{i,j} - \hat{R}_{i,j}| \leq \mathcal{D}_\Omega(\hat{R})$ is not necessary.
- The conclusion will be the same if $\mathcal{D}_\omega(\hat{R}) \leq \mathcal{D}_\Omega(\hat{R})$.

Proposition 1

- Let (a set of easily predictable entries) $\omega \subset \Omega(|\Omega| > 2)$ which satisfies that

$$\mathcal{D}_\omega(\hat{R}) \leq \mathcal{D}_\Omega(\hat{R})$$ ← the only change from **Theorem 1**

- Let (a set of entries hard to predict) $\Omega' = \Omega - \omega$.
- For any $\epsilon > 0$ *and* $1 > \lambda_0, \lambda_1 > 0$ *(*$\lambda_0 + \lambda_1 = 1$*),*
- $\lambda_0 \mathcal{D}_\Omega(\hat{R}) + \lambda_1 \mathcal{D}_{\Omega'}(\hat{R})$ *and* $\mathcal{D}_\Omega(\hat{R})$ are δ_1-stable and δ_2-stable respectively.
- Then, $\delta_1 \leq \delta_2$

How Many Entries Should We Remove?

- **Theorem 1** and **Proposition 1** only prove that it is beneficial to remove easily predictable entries from Ω.
- However, does not show **how many entries** we should remove.
- Theorem 2 shows that removing more entries that satisfy

$$|R_{i,j} - \hat{R}_{i,j}| \leq \mathcal{D}_{\Omega}(\hat{R})$$

can yield better Ω'.

Theorem 2

- Let $\omega_2 \subset \omega_1 \subset \Omega(|\Omega| > 2)$ satisfying $\forall (i,j) \in \omega_1, |R_{i,j} - \hat{R}_{i,j}| \leq \mathcal{D}_{\Omega}(\hat{R})$
- Let $\Omega_1 = \Omega - \omega_1$ and $\Omega_2 = \Omega - \omega_2$.
- For any $\epsilon > 0$ *and* $1 > \lambda_0, \lambda_1 > 0$ *(*$\lambda_0 + \lambda_1 = 1$*)*,
- $\lambda_0 \mathcal{D}_{\Omega}(\hat{R}) + \lambda_1 \mathcal{D}_{\Omega_1}(\hat{R})$ and $\lambda_0 \mathcal{D}_{\Omega}(\hat{R}) + \lambda_1 \mathcal{D}_{\Omega_2}(\hat{R})$ are δ_1-stable and δ_2-stable respectively.
- Then, $\delta_1 \leq \delta_2$

- Therefore, it is desirable to choose Ω' as the whole set of entries which are harder to predict than average: $\forall (i,j) \in \Omega', |R_{i,j} - \hat{R}_{i,j}| \geq \mathcal{D}_{\Omega}(\hat{R})$

Better Stability By Multiple Subsets

- **Theorem 1** and **Theorem 2** only consider choosing **one** Ω' to find stable matrix approximations.
- Is it possible to choose **more than one Ω'** that satisfy stable condition, and yield more stable solutions by minimizing them all together.
- The **Theorem 3** shows that incorporating *K* such entry sets (Ω') will be more stable than incorporating any (*K* − *1*) out of the *K* entry sets.

Theorem 3

- Let $\omega_1, \cdots, \omega_K \in \Omega(K > 1, |\Omega| > 2)$ satisfying
$$\forall (i,j) \in \omega_k \ (1 \leq k \leq K), |R_{i,j} - \hat{R}_{i,j}| \leq \mathcal{D}_\Omega(\hat{R})$$
- Let $\Omega_k = \Omega - \omega_k \ (\forall 1 \leq k \leq K)$.
- For any $\epsilon > 0$ and $1 > \lambda_0, \lambda_1, \cdots, \lambda_K > 0 \ \left(\Sigma_{i=0}^{K} \lambda_i = 1\right)$
- $\lambda_0 \mathcal{D}_\Omega(\hat{R}) + \sum_{k \in [1,K]} \lambda_k \mathcal{D}_{\Omega_k}(\hat{R})$ and $(\lambda_0 + \lambda_K)\mathcal{D}_\Omega(\hat{R}) + \sum_{k \in [1,K-1]} \lambda_k \mathcal{D}_{\Omega_k}(\hat{R})$ are δ_1-stable and δ_2-stable respectively.
- Then, $\delta_1 \leq \delta_2$

How to Find Multiple Subsets?

- **One of the Guidelines** to obtain hard predictable subsets of Ω?
  - 1) Choose $\omega \subset \Omega(|\Omega| > 2)$ satisfying $\forall (i,j) \in \omega, |R_{i,j} - \hat{R}_{i,j}| \leq \mathcal{D}_{\Omega}(\hat{R})$
    Let $\Omega_0 = \Omega - \omega$
  - 2) Divide ω into *K* non-overlapping subsets $\omega_1, \cdots, \omega_K$ with the condition that $\cup_{k \in [1,K]} \omega_k = \omega$, and let $\Omega_k = \Omega - \omega_k \ (\forall 1 \leq k \leq K)$
  - 3) Minimize $\lambda_0 \mathcal{D}_{\Omega}(\hat{R}) + \sum_{k=1}^{K} \lambda_k \mathcal{D}_{\Omega_k}(\hat{R})$

- Is it better than minimizing $\lambda_0 \mathcal{D}_{\Omega}(\hat{R}) + (1 - \lambda_0)\mathcal{D}_{\Omega_0}(\hat{R})$?
  - → **Theorem 4**

How to Find Multiple Subsets?

Theorem 4

- Let $\omega \subset \Omega(|\Omega| > 2)$ satisfying $\forall (i,j) \in \omega, |R_{i,j} - \hat{R}_{i,j}| \leq \mathcal{D}_{\Omega}(\hat{R})$
- Divide ω into K non-overlapping subsets $\omega_1, \cdots, \omega_K$ with the conditions that $\cup_{k \in [1,K]} \omega_k = \omega$
- Let $\Omega_0 = \Omega - \omega$ and $\Omega_k = \Omega - \omega_k$ $(\forall 1 \leq k \leq K)$.
- For any $\epsilon > 0$ and $1 > \lambda_0, \lambda_1, \cdots, \lambda_K > 0$ $\left(\Sigma_{i=0}^{K} \lambda_i = 1\right)$
- $\lambda_0 \mathcal{D}_{\Omega}(\hat{R}) + \sum_{k=1}^{K} \lambda_k \mathcal{D}_{\Omega_k}(\hat{R})$ and $\lambda_0 \mathcal{D}_{\Omega}(\hat{R}) + (1 - \lambda_0) \mathcal{D}_{\Omega_0}(\hat{R})$ are δ_1-stable and δ_2-stable respectively.
- Then, $\delta_1 \leq \delta_2$

Contents

KAIST

MLILAB
Machine Learning & Intelligence

Model Formulation

- Vanilla Low-rank Matrix Factorization:

$$\hat{R} = \arg\max_{X} \mathcal{D}_{\Omega}(X), rank(X) = r$$

- Model formulation of SMA:

$$\hat{R} = \arg\min_{X} \ \lambda_0 \mathcal{D}_{\Omega}(X) + \sum_{s=1}^{K} \lambda_s \mathcal{D}_{\Omega_s}(X) \quad s.t. \ rank(X) = r.$$

- By **Theorem 3**, Multiple hard predictable subsets are better than single predictable subsets. So, they make model can handle multiple subsets.
- Let $\Omega_1, \cdots, \Omega_K \in \Omega, \ \forall s \in [1, K], \mathcal{D}_{\Omega_x} \geq \mathcal{D}_{\Omega}$

Difficulties In Selecting Hard Predictable Subsets

- The more hard-predictable subsets the better stability.
- However, it is expensive to find such "hard predictable subsets". **Why?**
  - Because we don't know which subset of entries to choose without any prior knowledge.
  - We can't know final model without hard-predictable subsets. Also, we can't know hard-predictable subsets without final model. (Which came first, the chicken or the egg?)

- In other words,
  - Recall $\Omega_1, \cdots, \Omega_K \subset \Omega,\ \forall s \in [1, K], \mathcal{D}_{\Omega_s} \geq \mathcal{D}_{\Omega}$
  - To obtain such Ω_s is not trivial, because we can only check if the condition is satisfied with the final model.
  - But the final model cannot be known before we define and optimize a given loss function.

HARD PREDICTABLE SUBSETS SELECTION

- To address this issue:
  - 1) Approximate the targeted matrix *R* with existing LRMA solutions.
  - 2) for each entry $(i, j) \in \Omega$, it is chosen with large probability if and small probability otherwise
  - 3) Obtain Ω' by removing the chosen entries to satisfy the condition of **Proposition 1**, or probe Ω' to find hard predictable subsets that satisfy the condition of **Theorem 4**.
    - → In this paper they chose **Theorem 4**.

- Underlying Assumptions:
  - LRMF methods will not dramatically differ from final model of SMA $|R_{i,j} - \hat{R}_{i,j}| < \mathcal{D}_{\Omega}$
  - we can ensure that Ω' will satisfy $\mathcal{D}_{\Omega'} \geq \mathcal{D}_{\Omega}$ with high probability.

THE SMA ALGORITHM

Algorithm 1 The SMA Learning Algorithm

Require: R is the targeted matrix, Ω is the set of entries in R, and $\hat{R}$ is an approximation of R by existing L-RMA methods. $p > 0.5$ is the predefined probability for entry selection. μ_1 and μ_2 are the coefficients for L2-regularization.

$\Omega' = \emptyset$;
for each $(i, j) \in \Omega$ **do**
  randomly generate $\rho \in [0, 1]$;
  if $(|R_{i,j} - \hat{R}_{i,j}| \leq \mathcal{D}_\Omega \ \& \ \rho \leq p)$ or $(|R_{i,j} - \hat{R}_{i,j}| > \mathcal{D}_\Omega \ \& \ \rho \leq 1 - p)$ **then**
    $\Omega' \leftarrow \Omega' \cup \{(i, j)\}$;
  end if
end for
randomly divide Ω' into $\omega_1, ..., \omega_K$ $(\cup_{k=1}^{K} \omega_i = \Omega')$;
for all $k \in [1, K]$, $\Omega_k = \Omega - \omega_k$;
$(\hat{U}, \hat{V}) := \arg\min_{U,V} [\sum_{k=1}^{K} \lambda_k \mathcal{D}_{\Omega_k}(U^T V) + \lambda_0 \mathcal{D}_\Omega(UV^T) + \mu_1 \parallel U \parallel^2 + \mu_2 \parallel V \parallel^2]$
return $\hat{R} = \hat{U}\hat{V}^T$

Pretraining Part:
Finding Ω' for SMA algorithm
using **Theorem 4**

Core Training Part:
Add Regularizer to avoid overfitting

CONTENTS

KAIST

MLILAB
Machine Learning & Intelligence

Accuracy Comparisons

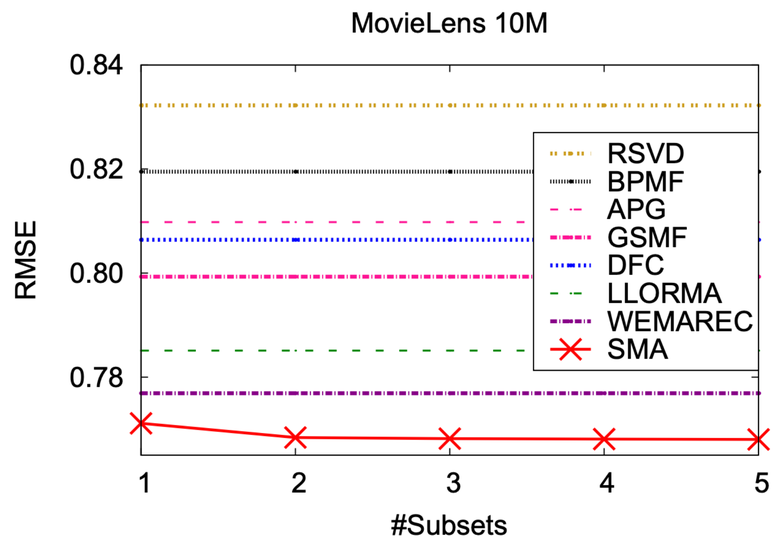
- Test RMSE on two benchmark datasets.

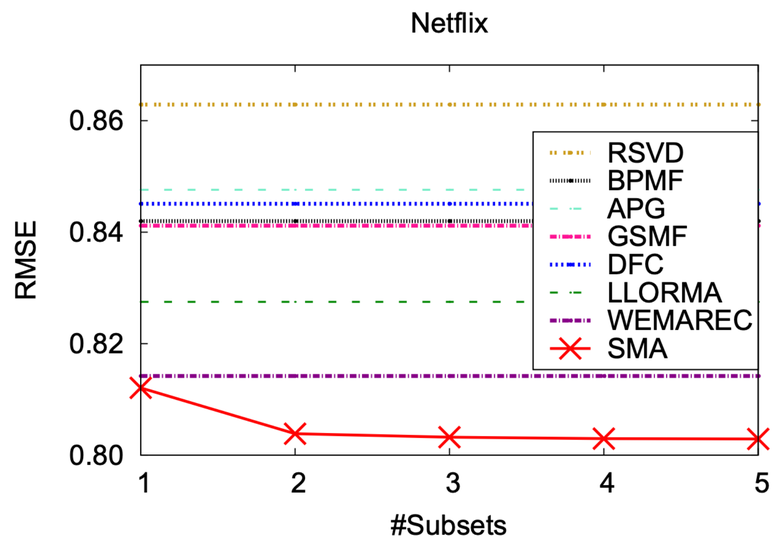
	MovieLens (10M)	Netflix
RSVD	0.8256 ± 0.0006	0.8534 ± 0.0001
BPMF	0.8197 ± 0.0004	0.8421 ± 0.0002
APG	0.8101 ± 0.0003	0.8476 ± 0.0003
GSMF	0.8012 ± 0.0011	0.8420 ± 0.0006
DFC	0.8067 ± 0.0002	0.8453 ± 0.0003
LLORMA	0.7855 ± 0.0002	0.8275 ± 0.0004
WEMAREC	0.7775 ± 0.0007	0.8143 ± 0.0001
SMA	$\mathbf{0.7682 \pm 0.0003}$	$\mathbf{0.8036 \pm 0.0004}$

- Compared to other state-of-the-arts models, it showed good performance.
- SMA demonstrates similar stability to ensembles such as DFC, LLORMA, and WEMAREC.
- Most importantly, other models can not guarantee a gap between training and test errors, but SMA is possible.

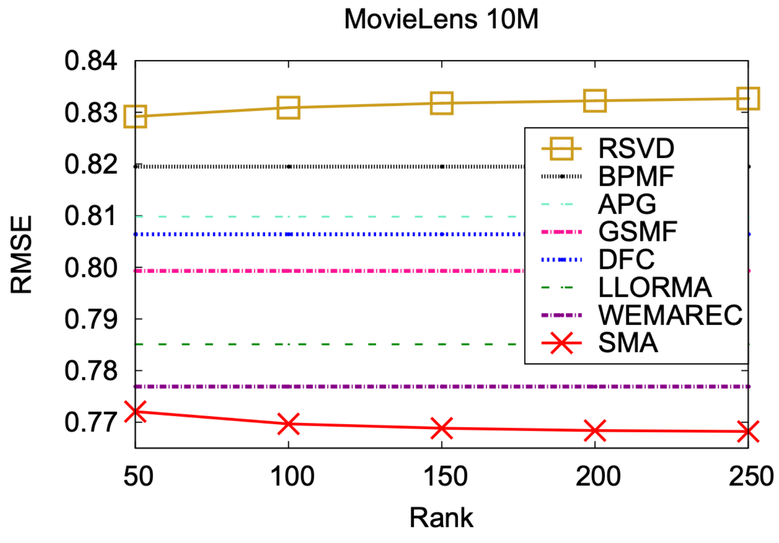
Effects of Hyper Parameters

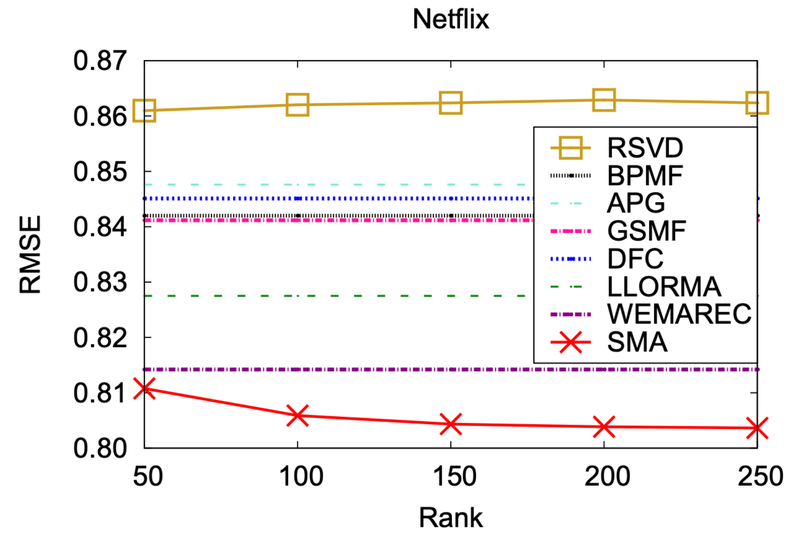
- RMSE vs the number of subsets *K*
  - As the number of subsets increases, the RMSE decreases.





- RMSE vs rank *r*
  - As the rank r increases, the SMA model does not overfit and the RMSE decreases.





Contents

Discussion Points

- Why did the authors choose **Theorem 4** rather than **Proposition 1** in the final algorithm?
  - I think we can make final algorithm with **Proposition 1**.

- The authors assumed that the model used in pretrain and the final model are similar. Is the random sampling steps are necessary?
  - I think they would better suggest reason why they use random sampling?

- Does this algorithm work for other fields or models?
  - There is no assumption related to low-rank matrix factorization model in **Theorem 1** or others. I think we can use this concept in other models.
  - I think this method can be regarded as regularization.

- Do we really need **Theorem 3**?
  - I think **Theorem 4** is enough for final SMA algorithm.

Any Questions?

KAIST