IEEE Data Science Workshop

Alternating Autoencoders for Matrix Completion

Kiwon Lee, Yong H. Lee, and Changho Suh
Korea Advanced Institute of Science and Technology (KAIST)

Overview

Autoencoders (AEs)

- Autoencoders (AEs) for matrix completion (MC) and collaborative filtering (CF)
- Item-based autoencoder (I-AE)

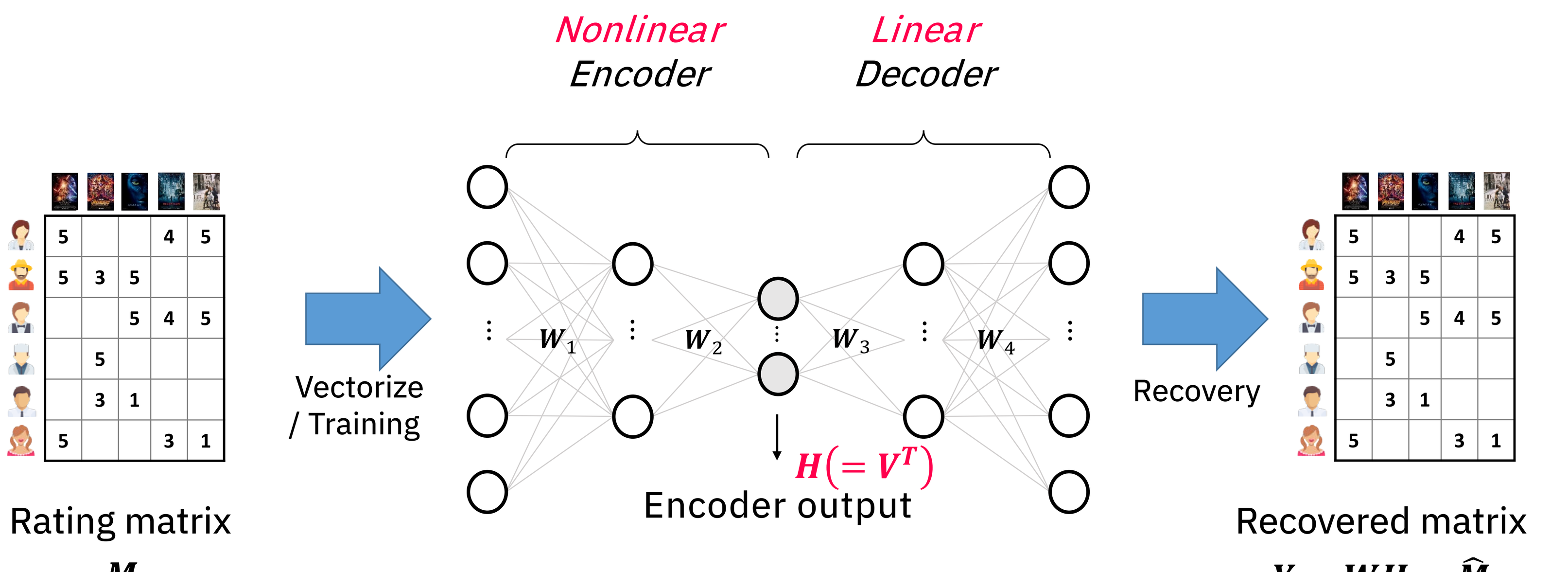

Fig. 1. Item-based autoencoder

- User-based autoencoder (U-AE)

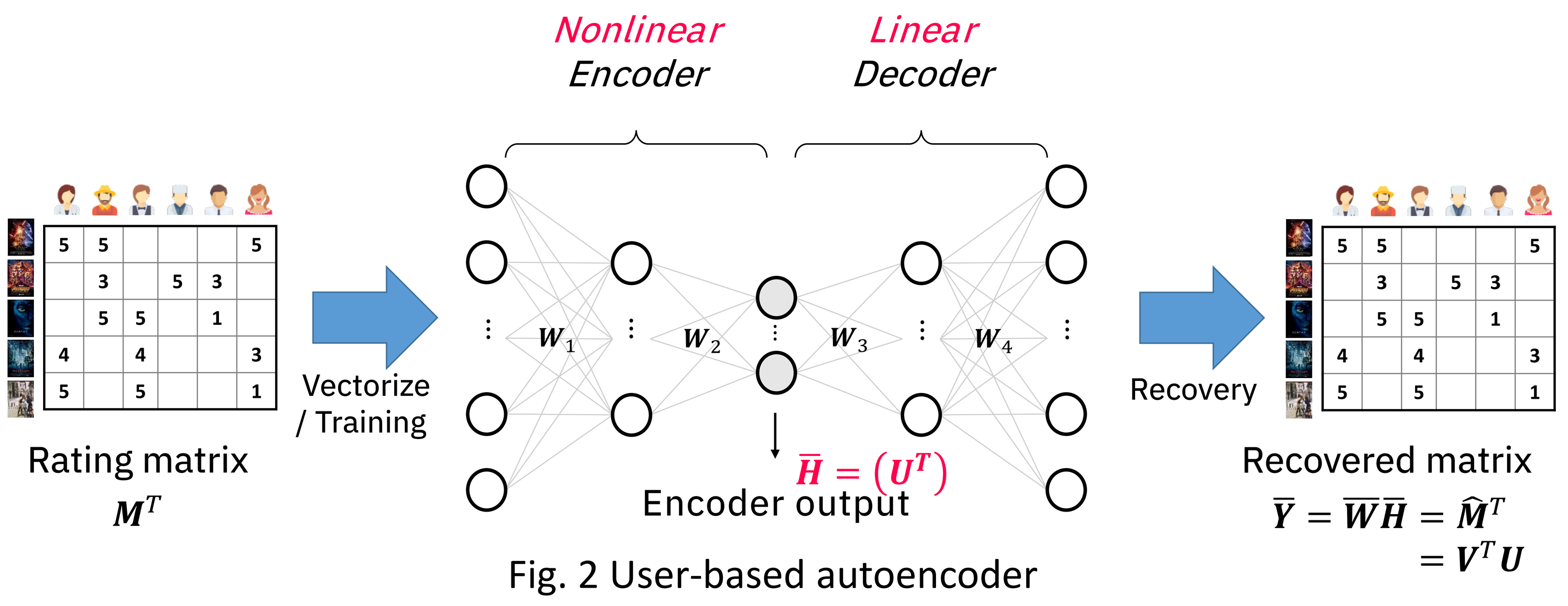

Fig. 2 User-based autoencoder

Sequential estimation property of AEs

- For I-AE, the encoder first estimates the item feature matrix $V^T(=H)$ and then the decoder estimates user feature matrix $U(=W)$ as a function of $V^T(=H)$.
- Similarly, for U-AE, $U^T(=\overline{H})$ is estimated first and then $V(=\overline{W})$ is obtained.
- This *sequential* estimation of feature matrices can cause some performance degradation, because one of the estimated feature matrices always depends on the other.

Alternating autoencoders

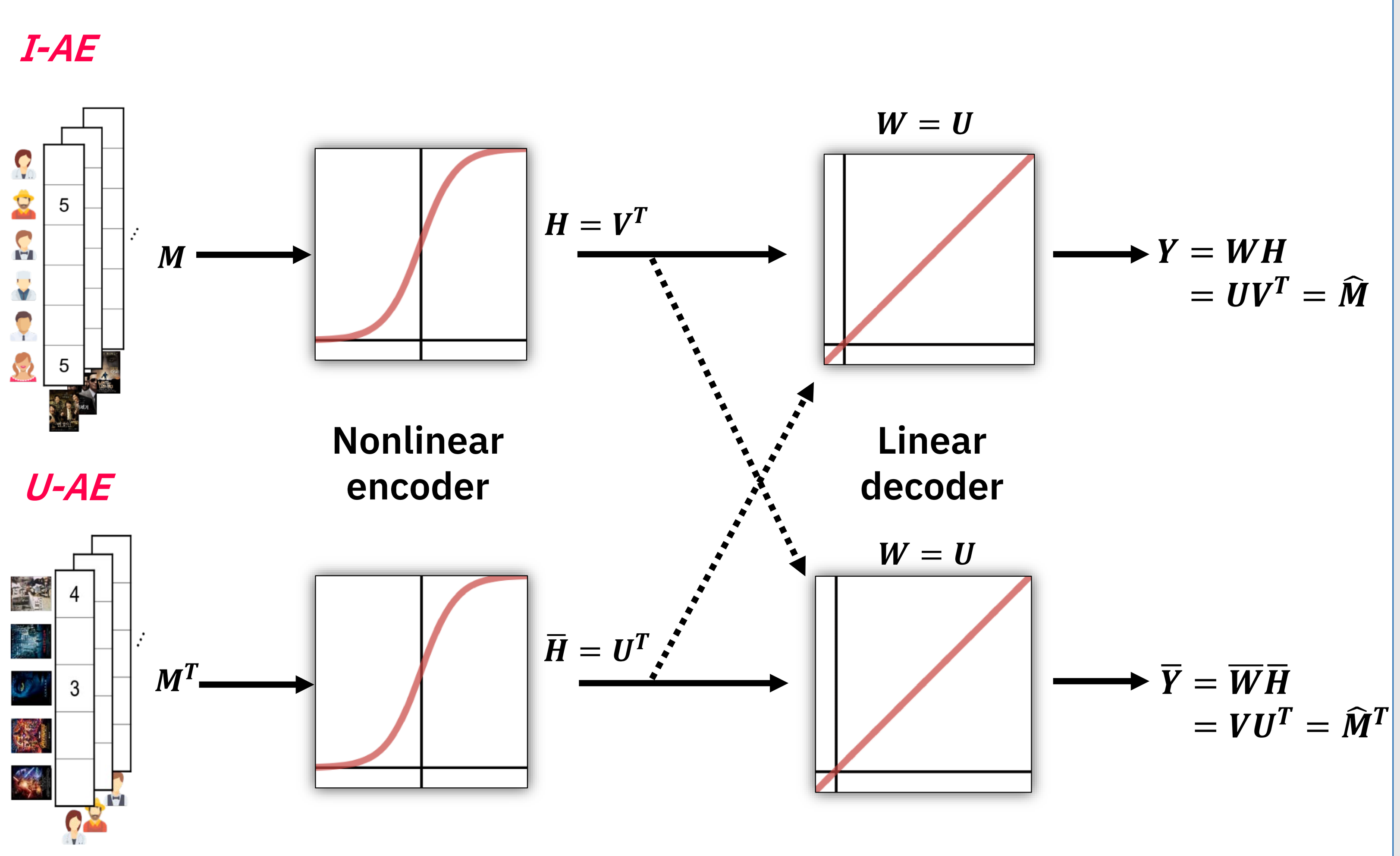

Fig. 3. Alternating autoencoder for matrix completion

- After each training epoch, AAE evaluates the RMSE of the current AE, and *alternates* with the other AE if the RMSE becomes less than the RMSE of the previous AE.

Alternating autoencoders

Algorithm 1 Alternating autoencoders.

1: **Input:** Incomplete matrix M, error tolerance δ and maximum training epoch $k_{\max}$.
2: **Initialize** $\overline{H}_0$ and all weighting matrices of AEs, $k = 0$, $\tau_0 = \infty$ and $k_{al} = 0$.
3: **repeat**
4: $k = k + 1$
5: Update H_k of I-AE, while fixing $W_k = \overline{H}_{k_{al}}^T$, via back-propagation.
6: Evaluate e_k.
7: **until** $e_k < \tau_{k-1}$
8: **Set** $\tau_k = e_k - \delta$ and $k_{al} = k$.
9: **repeat**
10: $k = k + 1$
11: Update $\overline{H}_k$ of U-AE, while fixing $\overline{W}_k = H_{k_{al}}^T$, via back-propagation.
12: Evaluate e_k.
13: **until** $e_k < \tau_k$
14: **Set** $\tau_k = e_k - \delta$ and $k_{al} = k$
15: **Go** to step 3 and repeat the process until $k = k_{\max}$.
16: **Output**: Estimated low-rank matrix $\widehat{M} = \overline{H}_{k_{\max}}^T H_{k_{\max}}$

Confirming sequential estimation property

Observation 1

- After convergence, the matrices Y, W and H for I-AE satisfy

$$\mathrm{rank}(H) = \mathrm{rank}(W) = \mathrm{rank}(Y)$$

- This observation is confirmed by showing that

$$\mathrm{col}(Y) = \mathrm{col}(W), \text{and } \mathrm{col}(Y^T) = \mathrm{col}(H^T)$$

via computer simulation.

Observation 2 (Matrix factorization)

$$Y = \widehat{M}, W = U, \text{and } H = V^T$$

- This is a consequence of Observation 1. It is also shown via simulation that each row of $W(=U)$ can be represented in terms of the pseudo-inverse of a submatrix of $H(=V^T)$

U-AE can be confirmed in a similar manner.

Experiments

Simulations with synthetic data

- Generate 500×500 matrices of rank 10, denoted as M_0
- Obtain incomplete matrices M by choosing $|M|$ entries of M_o
- Reconstruction rate: $\Pr(\text{reconstruction error} \leq 10^{-4})$

Simulations with practical data

- MovieLens-100k (943 users & 1,682 movies) MovieLens-1M (6,040 users & 3,952 movies)
- Train : Test = 9 : 1

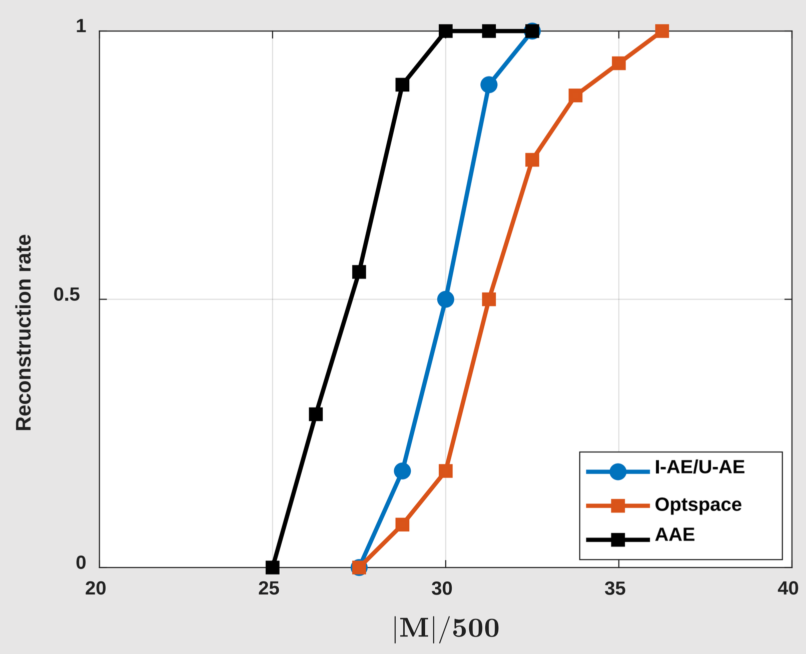

Fig. 4 Reconstruction rate

RMSE	ML-100k	ML-1M
Optspace	0.911	0.873
ALS-WR	0.913	0.843
LLORMA	0.898	0.833
CF-NADE	-	0.829
I-Autorec	-	0.831
U-AE	0.905	0.841
I-AE	0.884	0.829
AAE	**0.877**	**0.826**

Table 1. RMSE comparison

Contact

Kiwon Lee Department of EE, at KAIST; Email: kaiser5072@kaist.ac.kr; Phone: 82-010-2000-2954

IEEE Signal Processing Society