Personalized Machine Learning

Autoencoders for CF

Rodrigo Alves

October 23, 2025

Dimensionality Reduction

- Dimensionality reduction is a technique used to reduce the number of features in a dataset while retaining the most relevant information.
- Different types of data inputs, such as music, photos, or text, have unique characteristics that require specific machine learning approaches.
- Traditional methods like PCA may fail, particularly when dealing with data featuring non-linear relationships.
- Autoencoders, which are unsupervised neural networks, are employed for compressed data representation and are effective for dimensionality reduction and handling complex inputs.

Autoencoder

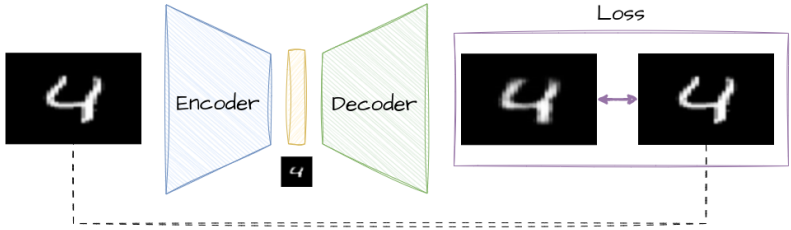
- An autoencoder is a type of feed-forward neural network.
- It is designed to reconstruct its input x_i as output x_i.
- Traditional methods like PCA may struggle, especially when dealing with non-linear relationships.
- To prevent trivial solutions, the network includes a bottleneck layer (or code layer) with significantly fewer dimensions than the input.

Autoencoder

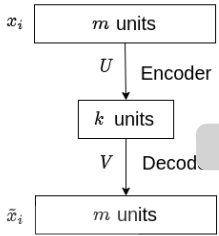
- An autoencoder is composed of both an encoder and a decoder.
- The encoder and the decoder typically have a similar structure.
- More formally, let $\mathcal{E}(x)$ be an encoder and $\mathcal{D}(x)$ be a decoder. Our optimization problem can be described as follows:

$$\min_{\mathcal{E},\mathcal{D}} \sum_{i} ||x_i - \mathcal{D}(\mathcal{E}(x_i))||$$

What is an autoencoder?



Autoencoders



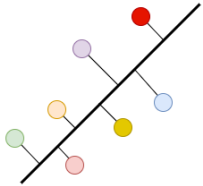
- The simplest autoencoder has one hidden layer and uses squared error loss.
- The optimization function can be denoted as:

$$\min_{U,V} \sum_i ||x_i - x_i UV||^2$$

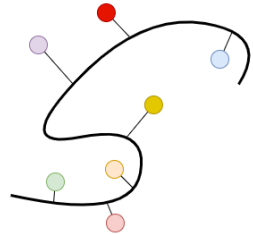
What happens if $k \geq m$?

- If $k < m$, then $UV \neq I$, where I represents the identity function.
- Therefore, if $k \geq m$, any simple solution where $UV = I$ is a trivial solution.
- More importantly, in the trivial case, there is no reduction of dimension.

PCA × Autoencoder

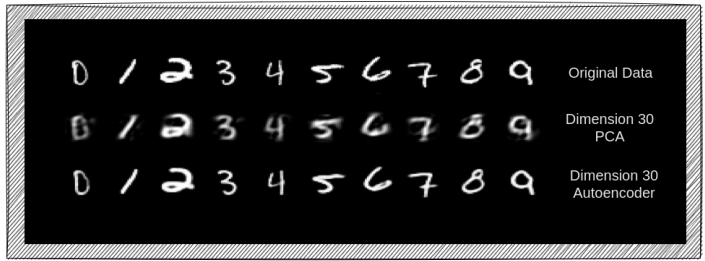


PCA



Autoencoder

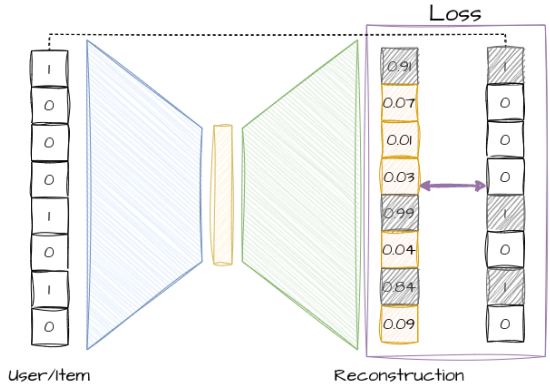
PCA × Autoencoder



Autoencoders for Collaborative Filtering

- Encode user-item interactions into a lower-dimensional space to discover meaningful patterns.
- Applicable to both explicit and implicit feedback.
- Effectively handle sparse user-item interaction data.
- Balancing model complexity and scalability, especially in large-scale recommendation systems.
- Potential for leveraging transfer learning to enhance content-based methods.

Autoencoder for implicit feedback



EASE

- EASE is the shallowest auto-encoder possible.
- It aims to solve the following problem:

$$\min_B ||X - XB||^2 + \lambda ||B||^2 \text{ s.t. } \text{diag}(B) = 0$$

- Why do we need the constraint $\text{diag}(B) = 0$?
- EASE has a closed-form solution, which we will explore shortly.
- Is this a good method?

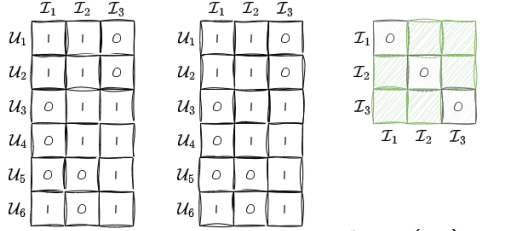
EASE Results

Table 1: Ranking accuracy (with standard errors of about 0.002, 0.001, and 0.001 on the *ML-20M*, *Netflix*, and *MSD* data, respectively), following the experimental set-up in [13].

(a) *ML-20M*	Recall@20	Recall@50	NDCG@100
popularity	0.162	0.235	0.191
EASE^R	0.391	0.521	0.420
EASE$^R_{\geq 0}$	0.373	0.499	0.402
results reproduced from [13]:			
SLIM	0.370	0.495	0.401
WMF	0.360	0.498	0.386
CDAE	0.391	0.523	0.418
MULT-VAE^{PR}	0.395	0.537	0.426
MULT-DAE	0.387	0.524	0.419
(b) *Netflix*			
popularity	0.116	0.175	0.159
EASE^R	0.362	0.445	0.393
EASE$^R_{\geq 0}$	0.345	0.424	0.373
results reproduced from [13]:			
SLIM	0.347	0.428	0.379
WMF	0.316	0.404	0.351
CDAE	0.343	0.428	0.376
MULT-VAE^{PR}	0.351	0.444	0.386
MULT-DAE	0.344	0.438	0.380
(c) *MSD*			
popularity	0.043	0.068	0.058
EASE^R	0.333	0.428	0.389
EASE$^R_{\geq 0}$	0.324	0.418	0.379
results reproduced from [13]:			
SLIM	— did not finish in [13] —		
WMF	0.211	0.312	0.257
CDAE	0.188	0.283	0.237
MULT-VAE^{PR}	0.266	0.364	0.316
MULT-DAE	0.266	0.363	0.313

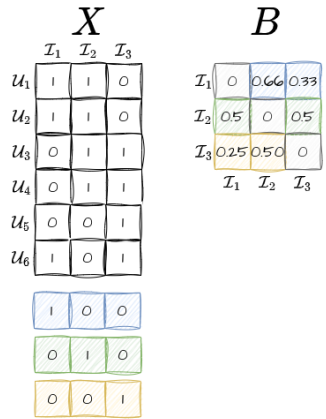
EASE: dimensions

$$|| X - X \times B ||^2_{\text{Fro}}$$



$$\text{diag}(B) = 0$$

EASE: intuition



EASE: closed-form

$$\min_B \|X - XB\|_F^2 + \lambda \|B\|_F^2$$
$$\text{s.t. } \text{diag(B)} = 0$$

Here $X \in \mathbb{R}^{m \times n}$ and $B \in \mathbb{R}^{n \times n}$.

Lagrangian:

$$\mathcal{L}(B) = \|X - XB\|_F^2 + \lambda \|B\|_F^2 + 2\gamma^\top \text{diag}(B)$$

EASE: closed-form

Lagrangian:

$$\mathcal{L}(B) = \|X - XB\|_F^2 + \lambda \|B\|_F^2 + 2\gamma^\top \text{diag}(B)$$

Derivative of the Lagrangian:

$$\begin{aligned}
\mathcal{L}'(B) &= 2(X)^\top (XB - X) + 2\lambda B + 2\text{diagMat}(\gamma) \\
&= 2X^\top (XB - X) + 2\lambda B + 2\text{diagMat}(\gamma) \\
&= 2X^\top XB - 2X^\top X + 2\lambda B + 2\text{diagMat}(\gamma)
\end{aligned}$$

EASE: closed-form

Derivative of the Lagrangian:

$$
\begin{aligned}
0 &= 2X^\top XB - 2X^\top X + 2\lambda B + 2\text{diagMat}(\gamma) \\
0 &= X^\top XB - X^\top X + \lambda B + \text{diagMat}(\gamma) \\
X^\top XB + \lambda B &= X^\top X - \text{diagMat}(\gamma) \\
(X^\top X + \lambda I)B &= X^\top X - \text{diagMat}(\gamma) \\
\hat{B} &= (X^\top X + \lambda I)^{-1}\big(X^\top X - \text{diagMat}(\gamma)\big)
\end{aligned}
$$

EASE: closed-form

Assume:

$$P = (X^\top X + \lambda I)^{-1}$$

We have:

$$\begin{aligned}
\hat{B} &= (X^\top X + \lambda I)^{-1}\big(X^\top X - \text{diagMat}(\gamma)\big) \\
&= (X^\top X + \lambda I)^{-1}\big(X^\top X + \lambda I - \lambda I - \text{diagMat}(\gamma)\big) \\
&= (X^\top X + \lambda I)^{-1}\big((X^\top X + \lambda I) - \lambda I - \text{diagMat}(\gamma)\big) \\
&= P\big(P^{-1} - \lambda I - \text{diagMat}(\gamma)\big) \\
&= I - P\big(\lambda I + \text{diagMat}(\gamma)\big)
\end{aligned}$$

EASE: closed-form </>

We know that:

$$\text{diag}(\hat{B}) = 0$$

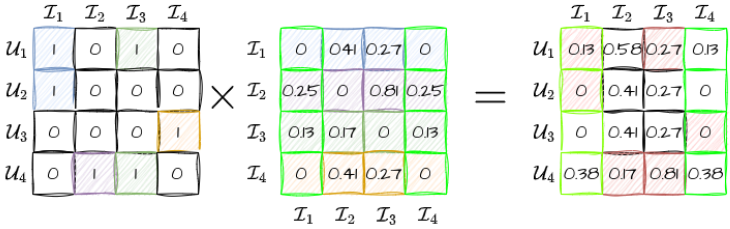
Therefore:

$$\begin{aligned} \text{diag}\Big(I - P\big(\lambda I + \text{diagMat}(\gamma)\big)\Big) &= \vec{0} \\ \text{diag}(I) - \text{diag}\big(P \times \text{diagMat}(\lambda \vec{1} + \gamma)\big) &= \vec{0} \end{aligned}$$

Finally:

$$\begin{aligned} \vec{1} - \text{diag}(P) \bigodot (\lambda \vec{1} + \gamma) &= \vec{0} \\ \vec{1} - \text{diag}(P) \bigodot \lambda \vec{1} - \text{diag}(P) \bigodot \gamma &= \vec{0} \\ \text{diag}(P) \bigodot \gamma &= \vec{1} - \lambda \text{diag}(P) \\ \gamma &= (\vec{1} - \lambda \text{diag}(P)) \oslash \text{diag}(P) \end{aligned}$$

Matrix $\hat{B}$


$\mathcal{I}_1$ $\mathcal{I}_2$ $\mathcal{I}_3$ $\mathcal{I}_4$
$\mathcal{U}_1$ 1 0 1 0
$\mathcal{U}_2$ 1 0 0 0
$\mathcal{U}_3$ 0 0 0 1
$\mathcal{U}_4$ 0 1 1 0
$\times$
$\mathcal{I}_1$ 0 0.41 0.27 0
$\mathcal{I}_2$ 0.25 0 0.81 0.25
$\mathcal{I}_3$ 0.13 0.17 0 0.13
$\mathcal{I}_4$ 0 0.41 0.27 0
$\mathcal{I}_1$ $\mathcal{I}_2$ $\mathcal{I}_3$ $\mathcal{I}_4$
$=$
$\mathcal{I}_1$ $\mathcal{I}_2$ $\mathcal{I}_3$ $\mathcal{I}_4$
$\mathcal{U}_1$ 0.13 0.58 0.27 0.13
$\mathcal{U}_2$ 0 0.41 0.27 0
$\mathcal{U}_3$ 0 0.41 0.27 0
$\mathcal{U}_4$ 0.38 0.17 0.81 0.38
$X_{[1:4,*]}$
B
$\hat{X}_{[1:4,*]}$

EASE: Scalability Limitation

We learned that EASE computes the item–item weight matrix $\hat{B}$ in closed form:

$$\hat{B} = P^{-1}(\text{diag}(P^{-1}))^{-1},$$

where $P = X^\top X + \lambda I$.

However:

- $P \in \mathbb{R}^{n \times n}$, where $n =$ number of items.
- We must **store and invert** this dense $n \times n$ matrix.
- Memory and time complexity $\mathcal{O}(n^2)$ and $\mathcal{O}(n^3)$ respectively.
- $\Rightarrow$ Not scalable when n is large (e.g., tens or hundreds of thousands of items).
- Oftentimes recommenders works with millions of users and items

ELSA: Scalable Linear Shallow Autoencoder

Idea: Replace the full item–item matrix $\hat{B}$ with a low-rank factorization.

$$\hat{B} = AA^\top, \quad A \in \mathbb{R}^{n \times r}, \quad r \ll n$$

Optimization objective:

$$\min_A \|X - X(AA^\top - I)\|_F^2$$

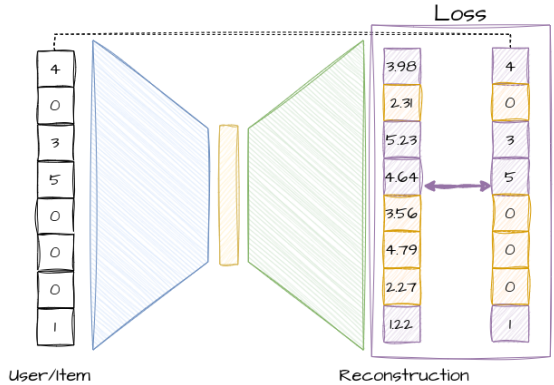
subject to:

$$\text{diag}(AA^\top) = 0, \quad \|A_{i,\cdot}\|_2 = 1$$

- Reduces parameters from n^2 to $n \times r$
- Trains via gradient descent (no matrix inversion)
- Keeps the same shallow linear structure as EASE

Result: Similar or better performance, with **linear scalability**.

What about explicit feedback? </>



Obrigado :) - Faculty of Information Technology