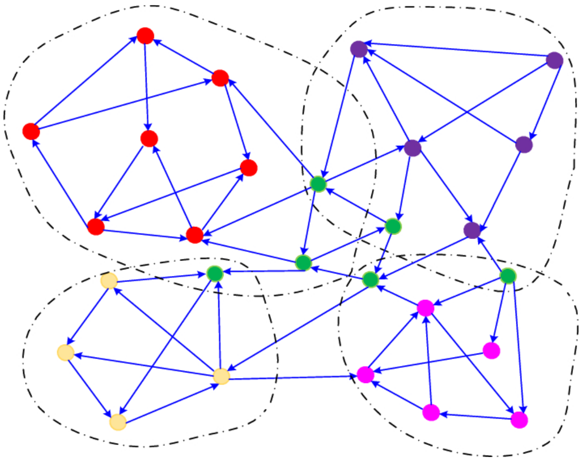
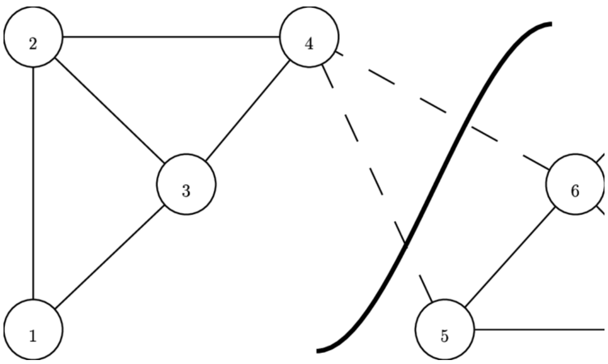
2.5 Graph Partitioning





Idea

- Divide graph into **k groups**
- Minimize edges between groups

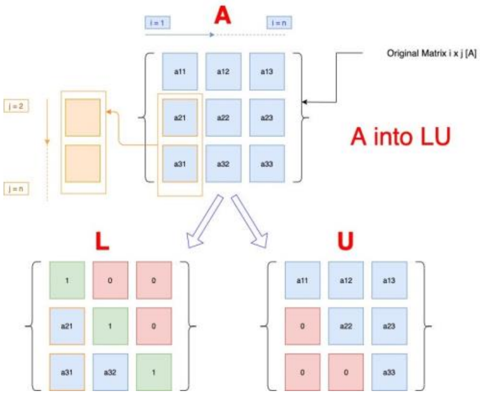
Example

- Divide students into study groups

Limitation

❌ Number of communities must be known in advance

Matrix Factorization



What is Matrix Factorization?

Matrix Factorization (MF) is a technique that decomposes a large matrix into smaller matrices whose product approximates the original matrix.

$$\mathbf{R} \approx \mathbf{P} \times \mathbf{Q}^T$$

Where:

- **R** → Original matrix (e.g., user–item ratings)
- **P** → User latent feature matrix
- **Q** → Item latent feature matrix

Why Matrix Factorization is Needed

Real-world matrices (especially in data science) are:

- Large
- Sparse
- High-dimensional

Matrix factorization helps to:
✓ Reduce dimensionality
✓ Discover hidden (latent) features
✓ Predict missing values

Basic Idea (Intuition)

Instead of storing full information, MF represents:

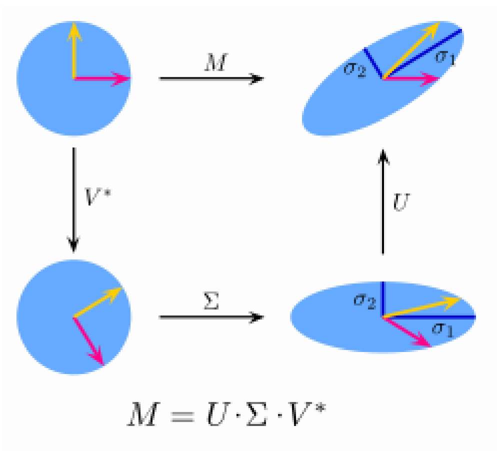
- Each **user** by a small set of preferences
- Each **item** by a small set of characteristics

Example:
A movie rating system may discover latent features like:

- Action vs Romance
- Comedy vs Drama

Types of Matrix Factorization

1. Singular Value Decomposition (SVD)



$$\therefore \Sigma = \begin{bmatrix} \sqrt{81} & 0 \\ 0 & \sqrt{1} \end{bmatrix} = \begin{bmatrix} 9 & 0 \\ 0 & 1 \end{bmatrix}$$

$$\therefore V = \begin{bmatrix} v_1, v_2 \end{bmatrix} = \begin{bmatrix} 0.447 & -0.894 \\ 0.894 & 0.447 \end{bmatrix}$$

U is found using formula $u_i = \frac{1}{\sigma_i} A \cdot v_i$

$$\therefore U = \begin{bmatrix} -0.894 & 0.447 \\ 0.447 & 0.894 \end{bmatrix}$$

$$A = U\Sigma V^T$$

- **U** → Left singular vectors
- **Σ** → Singular values (importance)
- **V** → Right singular vectors

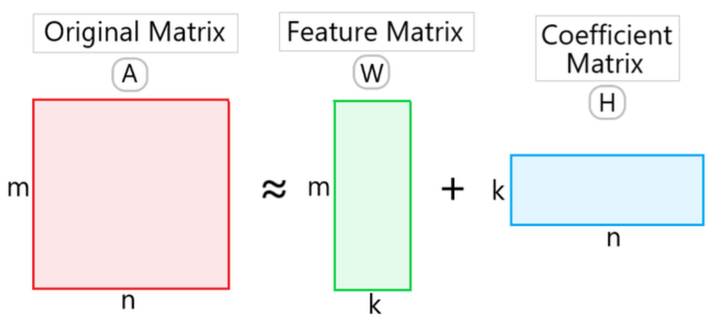
Applications

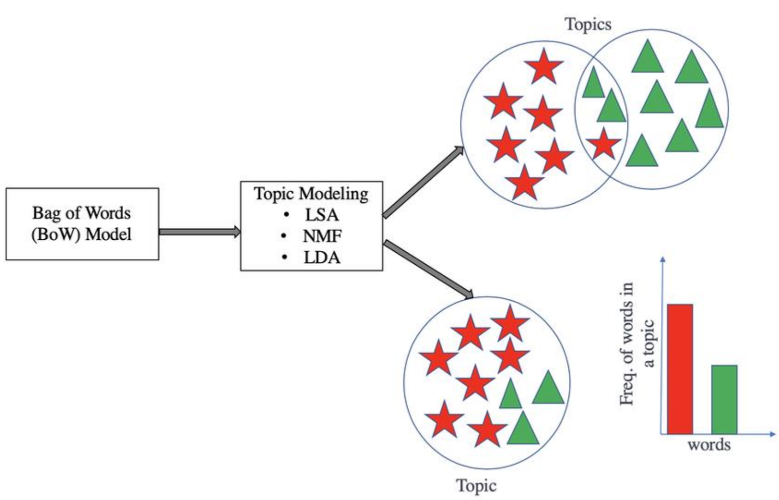
- Image compression
- Noise reduction
- Information retrieval

Limitation

❌ Works best on dense matrices

2. Non-Negative Matrix Factorization (NMF)





$$A \approx W \times H$$

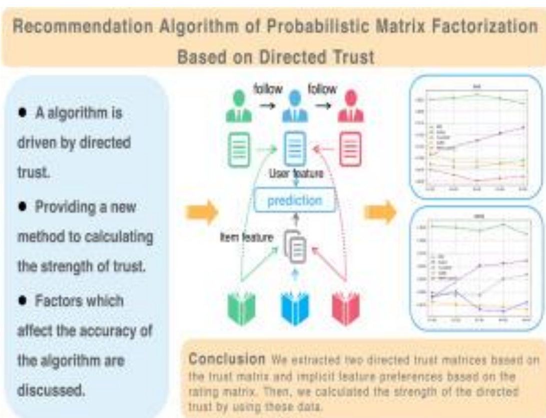
Key Feature

- All values are non-negative

Applications

- Topic modeling
- Text mining
- Image analysis

3. Probabilistic Matrix Factorization (PMF)



	Movie 1	Movie 2	Movie 3	Movie 4
	★★★			★★
		★★★	★	
	★★		★	★
			★★★	
		★★		★

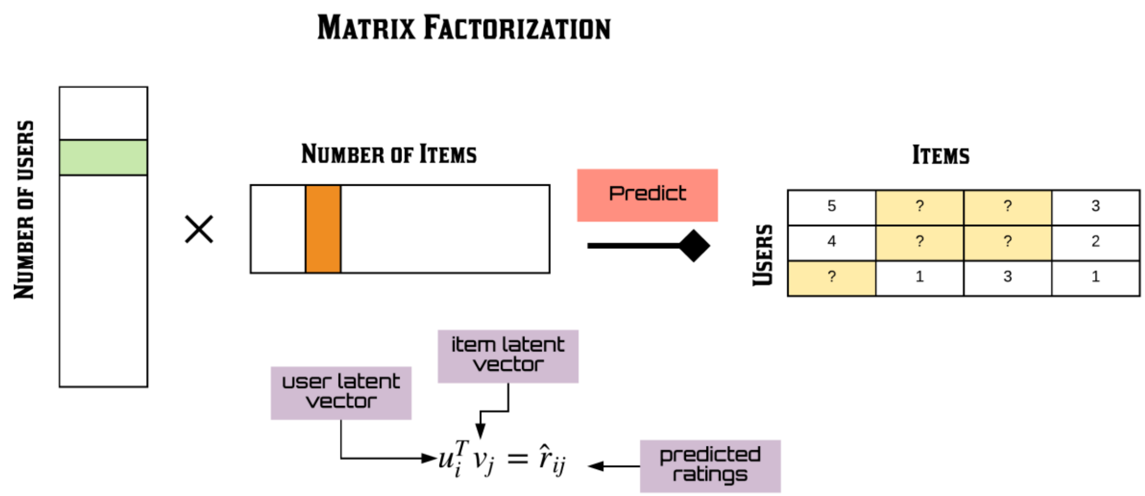
Idea

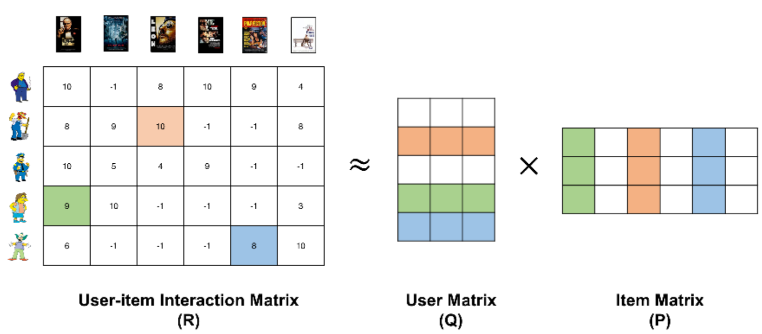
- Treats matrix entries as **random variables**
- Uses probability distributions

Application

- Recommendation systems

4. Matrix Factorization for Recommendation Systems





Used in Collaborative Filtering:

- Predicts missing ratings
- Learns user–item interactions

Prediction Formula

$$\hat{r}_{ui} = p_u^T q_i$$

Learning the Factors

Usually done using:

- Gradient Descent
- Alternating Least Squares (ALS)

Loss Function

$$\sum(r_{ui} - p_u^T q_i)^2 + \lambda(|| p_u ||^2 + || q_i ||^2)$$

Advantages

✓ Handles sparse data
✓ Scalable
✓ High prediction accuracy
✓ Interpretable latent features

Limitations

❌ Cold start problem
❌ Requires parameter tuning
❌ Hard to interpret features sometimes

Applications

✓ Recommendation systems (Netflix, Amazon)
✓ Image compression
✓ Topic modeling
✓ Signal processing