Data Pre-Processing-V

(Feature Extraction- SVD)

Dr. JASMEET SINGH

ASSISTANT PROFESSOR, CSED

TIET, PATIALA

Singular Valued Decomposition (SVD)

- In linear algebra, the singular value decomposition (SVD) is a factorization of a real or complex matrix.

- Formally, a matrix A of order m × n can be decomposed using SVD as follows:

$$A_{m\times n} = U_{m\times m}\Sigma_{m\times n}V^T_{n\times n}$$

- where U and V are column unit orthonormal vectors and Σ is a rectangular diagonal matrix whose diagonal entries are the singular values of matrix A.

- The number of non zero singular values is the rank of A.

Singular Valued Decomposition- Contd…

- U and V are orthonormal i.e.

$$UU^T = I \ or \ U^T = U^{-1}$$

$$VV^T = I \ or \ V^T = V^{-1}$$

- Singular values of any matrix $M_{m\times n}$ is the positive square root of the eigen values of matrix $M^T M$ of order n×n.

How to Compute U, Σ, and V ?

- Σ is a rectangular diagonal matrix of singular values of A.
- So, **in order to compute Σ, calculate eigen value of A^TA or AA^T i.e.**
  - Find λ's such that $|A^T A - \lambda I| = 0$
  - Compute positive square root of λ's to find singular values of A (say $\sigma_1, \sigma_2, \sigma_3, \dots \dots .. \sigma_n$) such that $\sigma_1 > \sigma_2 > \sigma_3 \dots\dots\dots > \sigma_n$
  - The diagonal entries of Σ is $(\sigma_1, \sigma_2, \sigma_3, \dots \dots .. \sigma_n)$ and rest all entries are 0.

How to Compute U, Σ, and V ?

- V is the column normalized eigen vectors of A^TA as explained below:

$$A^T A = (U\Sigma V^T)^T\ (U\Sigma V^T)$$
$$= V\Sigma^T U^T U\Sigma V^T$$
$$= V\Sigma\Sigma^T V^T \quad \text{(because U is orthonormal)}$$
$$= V\Sigma^2 V^T \quad \text{(because for diagonal matrix } AA^T=A^2\text{)}$$

Where, Σ^2 is the eigen value matrix of A^TA. So according to diagonalization process,

Therefore, V represents eigen vector of A^TA , since it is column unit vector so it must be normalized by each column.

How to Compute U, Σ, and V ?

- U is the column normalized eigen vectors of AA^T as explained below:

$$AA^T = (U\Sigma V^T)(U\Sigma V^T)^T$$

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

$$= U\Sigma\Sigma^T U^T \quad \text{(because V is orthonormal)}$$

$$= U\Sigma^2 U^T \quad \text{(because for diagonal matrix } AA^T = A^2\text{)}$$

Where, Σ^2 is the eigen value matrix of AA^T. So according to diagonalization process,

Therefore, U represents eigen vector of AA^T , since it is column unit vector so it must be normalized by each column.

How to Compute U, Σ, and V ?

- Alternatively, we can find U or V (anyone) using column normalized eigen vector of AA^T or A^TA respectively and then other can be found as

$$u_i = \frac{1}{\sigma_i} A v_i \text{ (because } AV = U\,\Sigma\text{)}$$

$$\text{or } v_i = \frac{1}{\sigma_i} A^T u_i \text{ (because } A^TU = V\,\Sigma\text{)}$$

SVD Example

Find the SVD of A, $U\Sigma V^T$, where

$$A = \begin{pmatrix} 3 & 2 & 2 \\ 2 & 3 & -2 \end{pmatrix}$$

SVD Example

Solution:

First we compute the singular values σ_i by finding the eigenvalues of AA^T

$$AA^T = \begin{pmatrix} 17 & 8 \\ 8 & 17 \end{pmatrix}$$

The characteristic polynomial is $\det(AA^T - \lambda I) = \lambda^2 - 34\lambda + 225 = (\lambda - 25)(\lambda - 9)$,

so the singular values are $\sigma_1 = \sqrt{25} = 5$ and $\sigma_2 = \sqrt{9} = 3$.

$$\text{Therefore } \Sigma = \begin{pmatrix} 5 & 0 & 0 \\ 0 & 3 & 0 \end{pmatrix}$$

In case, we will A^TA , we will have a 3X3 matrix and three values of λ which will be 25, 9, and 0.

SVD Example

- Now we find the columns of V by finding an orthonormal set of eigenvectors of A^TA. The eigenvalues of A^TA are 25, 9, and 0.
- For $\lambda = 25$, we have, $A^TA - 25I = \begin{pmatrix} -12 & 12 & 2 \\ 12 & -12 & -2 \\ 2 & -2 & -17 \end{pmatrix}$

The column normalized eigen vector of the above matrix is $v_1 = \begin{pmatrix} 1/\sqrt{2} \\ 1/\sqrt{2} \\ 0 \end{pmatrix}$

- For $\lambda = 9$, we have, $A^TA - 9I = \begin{pmatrix} 4 & 12 & 2 \\ 12 & 4 & -2 \\ 2 & -2 & -1 \end{pmatrix}$

The column normalized eigen vector of the above matrix is $v_2 = \begin{pmatrix} 1/\sqrt{18} \\ -1/\sqrt{18} \\ 4/\sqrt{18} \end{pmatrix}$

SVD Example

- For λ = 0, we have, $A^T A = \begin{pmatrix} 13 & 12 & 2 \\ 12 & 13 & -2 \\ 2 & -2 & 8 \end{pmatrix}$

The column normalized eigen vector of the above matrix is $v_2 = \begin{pmatrix} 2/3 \\ -2/3 \\ 1/3 \end{pmatrix}$

Therefore, $V = \begin{pmatrix} 1/\sqrt{2} & 1/\sqrt{18} & 2/3 \\ 1/\sqrt{2} & -1/\sqrt{18} & -2/3 \\ 0 & 4/\sqrt{18} & 1/3 \end{pmatrix}$

SVD Example

Finally, we can compute U by the formula $u_i = \frac{1}{\sigma_i} A v_i$

This gives $U = \begin{pmatrix} 1/\sqrt{2} & 1/\sqrt{2} \\ 1/\sqrt{2} & -1/\sqrt{2} \end{pmatrix}$

So in its full glory the SVD is:

$$A = U\Sigma V^T = \begin{pmatrix} 1/\sqrt{2} & 1/\sqrt{2} \\ 1/\sqrt{2} & -1/\sqrt{2} \end{pmatrix} \begin{pmatrix} 5 & 0 & 0 \\ 0 & 3 & 0 \end{pmatrix} \begin{pmatrix} 1/\sqrt{2} & 1/\sqrt{18} & 2/3 \\ 1/\sqrt{2} & -1/\sqrt{18} & -2/3 \\ 0 & 4/\sqrt{18} & 1/3 \end{pmatrix}^T$$

Relation between PCA and SVD

- Let the data matrix X be of $n \times p$ size, where n is the number of samples and p is the number of variables.
- Then the $p \times p$ covariance matrix C is a symmetric matrix and so it can be diagonalized:

$$C = VLV^T,$$

- where V is a matrix of eigenvectors (each column is an eigenvector) and L is a diagonal matrix with eigenvalues λi in the decreasing order on the diagonal.
- The eigenvectors are called *principal axes* or *principal directions* of the data.
- The coordinates of the i-th data point in the new PC space are given by the i-th row of XV.

Relation between PCA and SVD- Contd….

- If, we perform SVD on X we will get $X = U \sum V^T$
- If X is *centered*, i.e. column means have been subtracted and are now equal to zero, then the covariance matrix C is given by:

$$C = \frac{X^T X}{n-1} = \frac{V\Sigma^T U^T U \Sigma V^T}{n-1} = \frac{V\Sigma^2 V^T}{n-1} = V \frac{\Sigma^2}{n-1} V^T$$

- V are principal directions and that singular values are related to the eigenvalues of covariance matrix via $\lambda = \Sigma^2 /(n-1)$.
- Transformed dataset are given by $XV = U \sum V^T V = U \sum$

SVD for Dimensionality Reduction

- SVD is used for dimensionality reduction by using **compressed SVD**.
- In compressed SVD, dimensionality reduction is done by neglecting small singular values in the diagonal matrix Σ.
- In compressed SVD, the factorization has the form $U \Sigma V^T$. U is an m x p matrix. Σ is a p x p diagonal matrix. V is an n x p matrix, with V^T being the transpose of V, a p x n matrix, or the conjugate transpose if M contains complex values. The value p is called the rank.

Applications of SVD

- SVD, might be the most popular technique for dimensionality reduction when data is sparse.
- Sparse data refers to rows of data where many of the values are zero.
- This is often the case in some problem domains like recommender systems where a user has a rating for very few movies or songs in the database and zero ratings for all other cases.
- Another common example is a bag of words model of a text document, where the document has a count or frequency for some words and most words have a 0 value.

Applications of SVD

Examples of sparse data appropriate for applying SVD for dimensionality reduction:

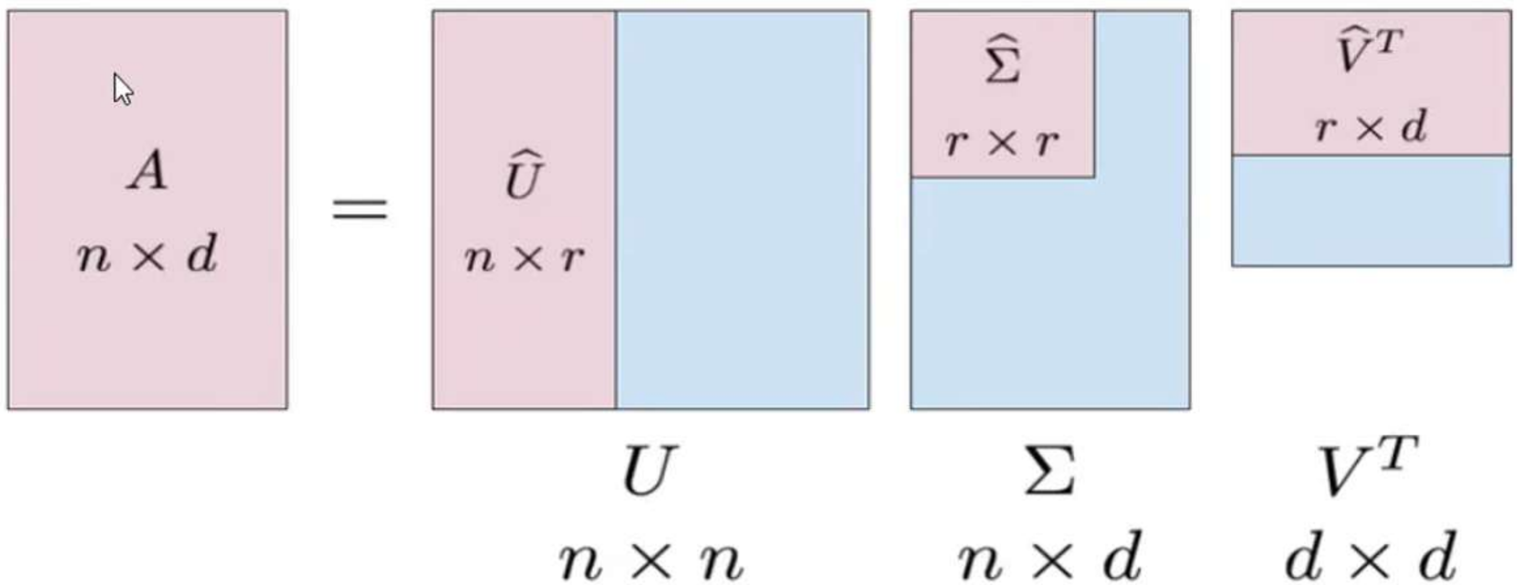
- Recommender Systems
- Customer-Product purchases
- User-Song Listen Counts
- User-Movie Ratings
- Text Classification
- One Hot Encoding
- Bag of Words Counts
- TF/IDF

Recommendation Systems and Singular Valued Decompositions

Singular Valued Decomposition-SVD

- SVD- a method from linear algebra that has been generally used as a dimensionality reduction technique in machine learning
- SVD is a matrix factorization technique, which reduces the number of features of a dataset by reducing the space dimension from N-dimension to K-dimension (where K<N)
- In RS, the SVD is used as a collaborative filtering technique
- It uses a matrix structure where each row represents a user, and each column represents an item
- The elements of this matrix are the ratings that are given to items by users

SVD for Recommender Systems



SVD for Recommender Systems

1. The Setup

In recommendation systems (like Netflix, Amazon, Spotify), we usually have a **user–item rating matrix (mXn)**:

- Rows = users, Columns = items (movies, products, songs); Entries = ratings (explicit like 1–5 stars, or implicit like clicks/views) Most of this matrix is **sparse** (missing ratings). We want to **predict the missing values** to make recommendations.

Example:

	Movie A	Movie B	Movie C	Movie D
User 1	5	?	3	?
User 2	?	4	?	2
User 3	2	?	?	5

SVD for Recommender Systems

2. Matrix Factorization Idea

SVD decomposes the rating matrix RRR into three matrices:

$$R \approx U\Sigma V^T$$

- U: User-feature matrix
- Σ: Singular values (importance of each feature)
- V: Item-feature matrix

Interpretation:

- Each user is represented by a vector of **latent features**.
- Each item (movie/product) is also represented by latent features.
- Ratings are approximated by the dot product of user and item features.

SVD for Recommender Systems

3. Low-Rank Approximation

We don’t need all singular values—only the top k largest ones (because data is noisy and sparse).

So we approximate:

$$R_k = U_k \Sigma_k V_k^T \ where\ k \ll \min(m, n)$$

This captures the **main patterns** (e.g., genres, taste preferences) while ignoring noise.

4. **Making Recommendations**

- Recommend the top-N items with highest predicted ratings that the user hasn’t seen yet.

SVD for Recommender Systems

- Besides making recommendations, SVD can also be used for **Cold-Start / Similarity Search.**
- Instead of just filling missing ratings, you can use the **latent representations** (from U_k and V_k) to:
  - Find **similar users** (nearest neighbors in latent space).
  - Find **similar items** (movies/products with similar latent vectors).
- This helps when a new user or item doesn't have many ratings.

SVD for Recommender Systems

- SVD can also be used for **Content Tagging & Categorization**
  - Latent features can serve as **automatic tags** for items.
  - Example: Suppose Movie X has high weight on “action” and “sci-fi” latent factors — SVD discovered this automatically, even without metadata.
- SVD can also be used for **Anomaly Detection**
  - If a user’s ratings strongly deviate from the low-rank SVD reconstruction, it may indicate:
    - Spam / fake reviews.
    - Anomalous behavior (like a hacked account).

Numerical Example

Step 1. User–item rating matrix R

Suppose we have **3 users** and **3 movies**. Ratings are from 1–5, with 0 meaning *not rated*:

$$R = \begin{bmatrix} 5 & 3 & 0 \\ 4 & 0 & 0 \\ 1 & 1 & 0 \end{bmatrix}$$

Numerical Example (Contd....)

Step 2. Apply SVD (truncated)

Decompose:

$$R \approx U \Sigma V^T$$

For simplicity, let's assume we keep **2 latent factors (k=2)** and compute (values rounded):

$$U = \begin{bmatrix} -0.82 & 0.27 \\ -0.57 & -0.73 \\ -0.07 & -0.62 \end{bmatrix}, \quad \Sigma = \begin{bmatrix} 7.06 & 0 \\ 0 & 2.57 \end{bmatrix}, \quad V^T = \begin{bmatrix} -0.78 & -0.63 & 0 \\ 0.62 & -0.77 & 0 \end{bmatrix}$$

Step 3. Reconstruct predicted ratings

$$\hat{R} = U \Sigma V^T$$

Gives (approx):

$$\hat{R} = \begin{bmatrix} 5.0 & 3.0 & 0.1 \\ 4.0 & 2.4 & 0.08 \\ 1.0 & .7 & 0.03 \end{bmatrix}$$

Numerical Example (Contd....)

Step 4. Interpret predictions

- Entry $\hat{R}_{ij}$ = predicted rating by user i for item j.
- Compare original vs predicted:

User	Movie 1	Movie 2	Movie 3
User 1	5 (known)	3 (known)	**0.1 (predicted)**
User 2	4 (known)	0 → **2.4 (predicted)**	0 → **0.08 (predicted)**
User 3	1 (known)	1 (known)	**0.03 (predicted)**

Step 5. Recommendations

- For **User 2**, Movie 2 has the highest predicted rating among unseen ones (2.4).
  👉 Recommend **Movie 2** to User 2.
- For **User 3**, Movie 3 has small predicted rating (0.03) → probably not worth recommending.

So the recommender would say:

- Suggest **Movie 2** to User 2.