Dimensionality Reduction

UV Decomposition
Singular-Value Decomposition
CUR Decomposition

Jeffrey D. Ullman
Stanford University
Often, our data can be represented by an $m$-by-$n$ matrix.

And this matrix can be closely approximated by the product of two matrices that share a small common dimension $r$. 

\[ M \approx U V \]
There are hidden, or *latent* factors that – to a close approximation – explain why the values are as they appear in the matrix.

Two kinds of data may exhibit this behavior:

1. Matrices representing a many-many-relationship.
   - “Latent” factors may explain the relationship.

2. Matrices that are really a relation (as in a relational database).
   - The columns may not really be independent.
Matrices as Relationships

- Our data can be a many-many relationship in the form of a matrix.
  - **Example**: people vs. movies; matrix entries are the ratings given to the movies by the people.
  - **Example**: students vs. courses; entries are the grades.

![Diagram](image)
Often, the relationship can be explained closely by *latent factors*.

- **Example**: genre of movies or books.
  - I.e., Joe liked Star Wars because Joe likes science-fiction, and Star Wars is a science-fiction movie.

- **Example**: types of courses.
  - Sue is good at computer science, and CS246 is a CS course.
Matrices as Relational Data

- Another closely related form of data is a collection of rows (tuples), each representing one entity.
- Columns represent attributes of these entities.
- **Example**: Stars can be represented by their mass, brightness in various color bands, diameter, and several other properties.
- But it turns out that there are only two independent variables (latent factors): mass and age.
<table>
<thead>
<tr>
<th>Star</th>
<th>Mass</th>
<th>Luminosity</th>
<th>Color</th>
<th>Age</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sun</td>
<td>1.0</td>
<td>1.0</td>
<td>Yellow</td>
<td>4.6B</td>
</tr>
<tr>
<td>Alpha Centauri</td>
<td>1.1</td>
<td>1.5</td>
<td>Yellow</td>
<td>5.8B</td>
</tr>
<tr>
<td>Sirius A</td>
<td>2.0</td>
<td>25</td>
<td>White</td>
<td>0.25B</td>
</tr>
</tbody>
</table>

The matrix
D-Dimensional Data Lying Close to a d-Dimensional Subspace

\[ D = 2 \]
\[ d = 1 \]

\[ D = 3 \]
\[ d = 2 \]
The axes of the subspace can be chosen by:

- The first dimension is the direction in which the points exhibit the greatest variance.
- The second dimension is the direction, orthogonal to the first, in which points show the greatest variance.
- And so on..., until you have enough dimensions that variance is really low.
The simplest form of matrix decomposition is to find a pair of matrixes, the first (U) with few columns and the second (V) with few rows, whose product is close to the given matrix M.

UV Decomposition

\[ M \approx U \times V \]
This decomposition works well if $r$ is the number of “hidden factors” that explain the matrix $M$.

**Example**: $m_{ij}$ is the rating person $i$ gives to movie $j$; $u_{ik}$ measures how much person $i$ likes genre $k$; $v_{kj}$ measures the extent to which movie $j$ belongs to genre $k$. 
Measuring the Error

- Common way to evaluate how well $P = UV$ approximates $M$ is by $RMSE$ (root-mean-square error).
- Average $(m_{ij} - p_{ij})^2$ over all $i$ and $j$.
- Take the square root.
  - Square-rooting changes the scale of error, but doesn’t affect which choice of $U$ and $V$ is best.
Example: RMSE

\[
\text{RMSE} = \sqrt{\frac{(0+0+1+0)}{4}} = \sqrt{0.25} = 0.5
\]

\[
\text{RMSE} = \sqrt{\frac{(0+0+0+4)}{4}} = \sqrt{1.0} = 1.0
\]

\text{Question for Thought: Are either of these the best choice?}
Optimizing U and V

- Pick $r$, the number of latent factors.
- Think of $U$ and $V$ as composed of variables, $u_{ik}$ and $v_{kj}$.
- Express the RMSE as (the square root of)
  \[ E = \sum_{ij} (m_{ij} - \sum_k u_{ik} v_{kj})^2. \]
- **Gradient descent**: repeatedly find the derivative of $E$ with respect to each variable and move each a small amount in the direction that lowers the value of $E$.

**Important point**: Go only a small distance, because $E$ is not linear, so following the derivative too far gets you off-course.
What if M is Missing Entries?

- Ignore the error term for $m_{ij}$ if that value is "unknown."
- **Example**: in a person-movie matrix, most movies are not rated by most people, so measure the error only for the known ratings.
  - To be covered by Jure in mid-February.
Expressions like this usually have many minima.
Seeking the nearest minimum from a starting point can trap you in a local minimum, from which no small improvement is possible.

But you can get trapped here

Global minimum
Avoiding Local Minima

- Use many different starting points, chosen at random, in the hope that one will be close enough to the global minimum.

- *Simulated annealing*: occasionally try a leap to someplace further away in the hope of getting out of the local trap.

  - **Intuition**: the global minimum might have many nearby local minima.
    - As Mt. Everest has most of the world’s tallest mountains in its vicinity.
Singular-Value Decomposition

Rank of a Matrix
Orthonormal Bases
Eigenvalues/Eigenvectors
Computing the Decomposition
Eliminating Dimensions
Why SVD?

- Gives a decomposition of any matrix into a product of three matrices.
- There are strong constraints on the form of each of these matrices.
  - Results in a decomposition that is essentially unique.
- From this decomposition, you can choose any number \( r \) of intermediate concepts (latent factors) in a way that minimizes the RMSE error given that value of \( r \).
The **rank** of a matrix is the maximum number of rows (or equivalently columns) that are linearly independent.

- I.e., no nontrivial sum is the all-zero vector.
  - **Trivial sum** = all coefficients are 0.

**Example:** Exist two independent rows.

- In fact, no row is a multiple of another in this example.

**But any** 3 rows are dependent.

- **Example:** First + third – twice the second = [0,0,0].

Similarly, the 3 columns are dependent.

Therefore, rank = 2.
Important Fact About Rank

- If a matrix has rank $r$, then it can be decomposed exactly into matrices whose shared dimension is $r$.

- **Example**, in Sect. 11.3 of MMDS, of a 7-by-5 matrix with rank 2 and an exact decomposition into a 7-by-2 and a 2-by-5 matrix.
Vectors are orthogonal if their dot product is 0.

Example: \([1,2,3].[1,-2,1] = 1*1 + 2*(-2) + 3*1 = 1-4+3 = 0\), so these two vectors are orthogonal.

A unit vector is one whose length is 1.

- Length = square root of sum of squares of components.
  - No need to take square root if we are looking for length = 1.

Example: \([0.8, -0.1, 0.5, -0.3, 0.1]\) is a unit vector, since \(0.64 + 0.01 + 0.25 + 0.09 + 0.01 = 1\).

An orthonormal basis is a set of unit vectors any two of which are orthogonal.
Example: Columns Are Orthonormal

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>$3/\sqrt{116}$</td>
<td>$1/2$</td>
<td>$7/\sqrt{116}$</td>
<td>$1/2$</td>
</tr>
<tr>
<td>$3/\sqrt{116}$</td>
<td>$-1/2$</td>
<td>$7/\sqrt{116}$</td>
<td>$-1/2$</td>
</tr>
<tr>
<td>$7/\sqrt{116}$</td>
<td>$1/2$</td>
<td>$-3/\sqrt{116}$</td>
<td>$-1/2$</td>
</tr>
<tr>
<td>$7/\sqrt{116}$</td>
<td>$-1/2$</td>
<td>$-3/\sqrt{116}$</td>
<td>$1/2$</td>
</tr>
</tbody>
</table>
Form of SVD

Special conditions:

- $U$ and $V$ are column-orthonormal
- (so $V^T$ has orthonormal rows)
- $\Sigma$ is a diagonal matrix
The values of $\Sigma$ along the diagonal are called the *singular values*.

It is always possible to decompose $M$ exactly, if $r$ is the rank of $M$.

But usually, we want to make $r$ much smaller than the rank, and we do so by setting to 0 the smallest singular values.

- Which has the effect of making the corresponding columns of $U$ and $V$ useless, so they may as well not be there.
Linkage Among Components of $U, V, \Sigma$

$$A \approx U\Sigma V^T = \sum_i \sigma_i u_i \circ v_i^T$$
Each Singular Value Affects One Column of U and V

\[ A \approx U \Sigma V^T = \sum_i \sigma_i u_i \circ v_i^T \]

If we set \( \sigma_2 = 0 \), then the green columns may as well not exist.
The following is Example 11.9 from MMDS.
It modifies the simpler Example 11.8, where a rank-2 matrix can be decomposed exactly into a 7-by-2 $U$ and a 5-by-2 $V$. 
Example: Users-to-Movies

\[ A = U \Sigma V^T \] - example: Users to Movies

\[
\begin{bmatrix}
1 & 1 & 1 & 0 & 0 \\
3 & 3 & 3 & 0 & 0 \\
4 & 4 & 4 & 0 & 0 \\
5 & 5 & 5 & 0 & 0 \\
0 & 2 & 0 & 4 & 4 \\
0 & 0 & 0 & 5 & 5 \\
0 & 1 & 0 & 2 & 2
\end{bmatrix}
\begin{bmatrix}
0.13 & 0.02 & -0.01 \\
0.41 & 0.07 & -0.03 \\
0.55 & 0.09 & -0.04 \\
0.68 & 0.11 & -0.05 \\
0.15 & -0.59 & 0.65 \\
0.07 & -0.73 & -0.67 \\
0.07 & -0.29 & 0.32
\end{bmatrix}
\begin{bmatrix}
12.4 & 0 & 0 \\
0 & 9.5 & 0 \\
0 & 0 & 1.3
\end{bmatrix}
\begin{bmatrix}
0.56 & 0.59 & 0.56 & 0.09 & 0.09 \\
0.12 & -0.02 & 0.12 & -0.69 & -0.69 \\
0.40 & -0.80 & 0.40 & 0.09 & 0.09
\end{bmatrix}
\]
**Example: Users-to-Movies**

\[
A = U \Sigma V^T - \text{example: Users to Movies}
\]

<table>
<thead>
<tr>
<th>SciFi-concept</th>
<th>Romance-concept</th>
</tr>
</thead>
<tbody>
<tr>
<td>SciFi</td>
<td>Romance</td>
</tr>
<tr>
<td>Alien</td>
<td>Serenity</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

\[
\begin{bmatrix}
1 & 1 & 1 & 0 & 0 \\
3 & 3 & 3 & 0 & 0 \\
4 & 4 & 4 & 0 & 0 \\
5 & 5 & 5 & 0 & 0 \\
0 & 2 & 0 & 4 & 4 \\
0 & 0 & 0 & 5 & 5 \\
0 & 1 & 0 & 2 & 2
\end{bmatrix}
= 
\begin{bmatrix}
0.13 & 0.02 & -0.01 \\
0.41 & 0.07 & -0.03 \\
0.55 & 0.09 & -0.04 \\
0.68 & 0.11 & -0.05 \\
0.15 & -0.59 & 0.65 \\
0.07 & -0.73 & -0.67 \\
0.07 & -0.29 & 0.32
\end{bmatrix}
\begin{bmatrix}
12.4 & 0 & 0 \\
0 & 9.5 & 0 \\
0 & 0 & 1.3
\end{bmatrix}
\begin{bmatrix}
0.56 & 0.59 & 0.56 & 0.09 & 0.09 \\
0.12 & -0.02 & 0.12 & -0.69 & -0.69 \\
0.40 & -0.80 & 0.40 & 0.09 & 0.09
\end{bmatrix}
\]
Example: Users-to-Movies

\[ A = U \Sigma V^T \text{ - example:} \]

- **A** is the matrix representing the users-to-movies relationship.
- **U** is the "user-to-concept" similarity matrix.

\[
\begin{bmatrix}
1 & 1 & 1 & 0 & 0 \\
3 & 3 & 3 & 0 & 0 \\
4 & 4 & 4 & 0 & 0 \\
5 & 5 & 5 & 0 & 0 \\
0 & 2 & 0 & 4 & 4 \\
0 & 0 & 0 & 5 & 5 \\
0 & 1 & 0 & 2 & 2
\end{bmatrix}
\]

\[
\begin{bmatrix}
0.13 & 0.02 & -0.01 \\
0.41 & 0.07 & -0.03 \\
0.55 & 0.09 & -0.04 \\
0.68 & 0.11 & -0.05 \\
0.15 & -0.59 & 0.65 \\
0.07 & -0.73 & -0.67 \\
0.07 & -0.29 & 0.32
\end{bmatrix}
\]

\[ U \text{ is "user-to-concept" similarity matrix} \]
### Example: Users-to-Movies

\[ A = U \Sigma V^T \] - example:

<table>
<thead>
<tr>
<th>SciFi</th>
<th>Romance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Matrix</td>
<td>Alien</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

SciFi-concept

\[
\begin{bmatrix}
0.13 & 0.02 & -0.01 \\
0.41 & 0.07 & -0.03 \\
0.55 & 0.09 & -0.04 \\
0.68 & 0.11 & -0.05 \\
0.15 & -0.59 & 0.65 \\
0.07 & -0.73 & -0.67 \\
0.07 & -0.29 & 0.32 \\
\end{bmatrix}
\]

“strength” of the SciFi-concept

\[
\begin{bmatrix}
12.4 & 0 & 0 \\
0 & 9.5 & 0 \\
0 & 0 & 1.3 \\
\end{bmatrix}
\]
Example: Users-to-Movies

\[ A = U \Sigma V^T \] - example:

\[ V \] is “movie-to-concept” similarity matrix.

\[
\begin{pmatrix}
1 & 1 & 1 & 0 & 0 \\
3 & 3 & 3 & 0 & 0 \\
4 & 4 & 4 & 0 & 0 \\
5 & 5 & 5 & 0 & 0 \\
0 & 2 & 0 & 4 & 4 \\
0 & 0 & 0 & 5 & 5 \\
0 & 1 & 0 & 2 & 2
\end{pmatrix}
\]

\[
\begin{pmatrix}
0.13 & 0.02 & -0.01 \\
0.41 & 0.07 & -0.03 \\
0.55 & 0.09 & -0.04 \\
0.68 & 0.11 & -0.05 \\
0.15 & -0.59 & 0.65 \\
0.07 & -0.73 & -0.67 \\
0.07 & -0.29 & 0.32
\end{pmatrix}
\]

\[
\begin{pmatrix}
12.4 & 0 & 0 \\
0 & 9.5 & 0 \\
0 & 0 & 1.3
\end{pmatrix}
\]
Q: How exactly is dimensionality reduction done?
A: Set smallest singular values to zero
Q: How exactly is dimensionality reduction done?
A: Set smallest singular values to zero
How exactly is dimensionality reduction done?

Set smallest singular values to zero.
**Lowering the Dimension**

- **Q:** How exactly is dimensionality reduction done?
- **A:** Set smallest singular values to zero

$$
\begin{bmatrix}
1 & 1 & 1 & 0 & 0 \\
3 & 3 & 3 & 0 & 0 \\
4 & 4 & 4 & 0 & 0 \\
5 & 5 & 5 & 0 & 0 \\
0 & 2 & 0 & 4 & 4 \\
0 & 0 & 0 & 5 & 5 \\
0 & 1 & 0 & 2 & 2 \\
\end{bmatrix} \approx \begin{bmatrix}
0.92 & 0.95 & 0.92 & 0.01 & 0.01 \\
2.91 & 3.01 & 2.91 & -0.01 & -0.01 \\
3.90 & 4.04 & 3.90 & 0.01 & 0.01 \\
4.82 & 5.00 & 4.82 & 0.03 & 0.03 \\
0.70 & 0.53 & 0.70 & 4.11 & 4.11 \\
-0.69 & 1.34 & -0.69 & 4.78 & 4.78 \\
0.32 & 0.23 & 0.32 & 2.01 & 2.01 \\
\end{bmatrix}
$$
Frobenius Norm and Approximation Error

- The *Frobenius norm* of a matrix is the square root of the sum of the squares of its elements.
- The *error* in an approximation of one matrix by another is the Frobenius norm of the difference.
  - Same as the RMSE.
- **Important fact**: The error in the approximation of a matrix by SVD, subject to picking $r$ singular values, is minimized by zeroing all but the largest $r$ singular values.
So what’s a good value for r?
Let the *energy* of a set of singular values be the sum of their squares.
Pick r so the retained singular values have at least 90% of the total energy.
**Example:** With singular values 12.4, 9.5, and 1.3, total energy = 245.7.
If we drop 1.3, whose square is only 1.7, we are left with energy 244, or over 99% of the total.
But also dropping 9.5 leaves us with too little.
We want to describe how the SVD is actually computed.

Essential is a method for finding the principal eigenvalue (the largest one) and the corresponding eigenvector of a symmetric matrix.

- M is symmetric if $m_{ij} = m_{ji}$ for all i and j.

Start with any “guess eigenvector” $x_0$.

Construct $x_{k+1} = Mx_k / ||Mx_k||$ for $k = 0, 1, ...$

- $||...||$ denotes the Frobenius norm.

Stop when consecutive $x_k$'s show little change.
Example: Iterative Eigenvector

\[
M = \begin{pmatrix}
1 & 2 \\
2 & 3 \\
\end{pmatrix} \quad x_0 = \begin{pmatrix}
1 \\
1 \\
\end{pmatrix}
\]

\[
\frac{Mx_0}{||Mx_0||} = \frac{3}{5} \sqrt{34} = \frac{0.51}{0.86} = x_1
\]

\[
\frac{Mx_1}{||Mx_1||} = \frac{2.23}{3.60} \sqrt{17.93} = \frac{0.53}{0.85} = x_2
\]
Once you have the principal eigenvector $\mathbf{x}$, you find its eigenvalue $\lambda$ by $\lambda = \mathbf{x}^T \mathbf{M} \mathbf{x}$.

**In proof:** we know $\mathbf{x}\lambda = \mathbf{M}\mathbf{x}$ if $\lambda$ is the eigenvalue; multiply both sides by $\mathbf{x}^T$ on the left.

- Since $\mathbf{x}^T \mathbf{x} = 1$ we have $\lambda = \mathbf{x}^T \mathbf{M} \mathbf{x}$.

**Example:** If we take $\mathbf{x}^T = [0.53, 0.85]$, then $\lambda =$

$$
[0.53 \ 0.85] \begin{bmatrix} 1 & 2 \\ 2 & 3 \end{bmatrix} \begin{bmatrix} 0.53 \\ 0.85 \end{bmatrix} = 4.25
$$
Eliminate the portion of the matrix $M$ that can be generated by the first eigenpair, $\lambda$ and $x$.

$M^* := M - \lambda x x^T$.

Recursively find the principal eigenpair for $M^*$, eliminate the effect of that pair, and so on.

**Example:**

$$M^* = \begin{bmatrix} 1 & 2 \\ 2 & 3 \end{bmatrix} - 4.25 \begin{bmatrix} 0.53 \\ 0.85 \end{bmatrix} \begin{bmatrix} 0.53 \\ 0.85 \end{bmatrix} = \begin{bmatrix} -0.19 & 0.09 \\ 0.09 & 0.07 \end{bmatrix}$$
How to Compute the SVD

- Start by supposing $M = U\Sigma V^T$.
- $M^T = (U\Sigma V^T)^T = (V^T)^T \Sigma^T U^T = V \Sigma U^T$.
  - Why? (1) Rule for transpose of a product (2) the transpose of the transpose and the transpose of a diagonal matrix are both the identity function.
- $M^T M = V \Sigma U^T U \Sigma V^T = V \Sigma^2 V^T$.
  - Why? $U$ is orthonormal, so $U^T U$ is an identity matrix.
  - Also note that $\Sigma^2$ is a diagonal matrix whose i-th element is the square of the i-th element of $\Sigma$.
- $M^T M V = V \Sigma^2 V^T V = V \Sigma^2$.
  - Why? $V$ is also orthonormal.
Starting with \((M^T M)V = V\Sigma^2\), note that therefore the i-th column of \(V\) is an eigenvector of \(M^T M\), and its eigenvalue is the i-th element of \(\Sigma^2\).

Thus, we can find \(V\) and \(\Sigma\) by finding the eigenpairs for \(M^T M\).

- Once we have the eigenvalues in \(\Sigma^2\), we can find the singular values by taking the square root of these eigenvalues.

Symmetric argument, starting with \(MM^T\), gives us \(U\).
CUR Decomposition

The Sparsity Issue
Picking Random Rows and Columns
It is common for the matrix $M$ that we wish to decompose to be very sparse. But $U$ and $V$ from a UV or SVD decomposition will not be sparse even so. CUR decomposition solves this problem by using only (randomly chosen) rows and columns of $M$.
Form of CUR Decomposition

M = randomly chosen columns of M.
C = randomly chosen columns of M.
R = randomly chosen rows of M
U is tricky – more about this.
Construction of U

- U is r-by-r, so it is small, and it is OK if it is dense and complex to compute.
- Start with \( W = \) intersection of the r columns chosen for C and the r rows chosen for R.
- Compute the SVD of \( W \) to be \( X\Sigma Y^T \).
- Compute \( \Sigma^+ \), the *Moore-Penrose inverse* of \( \Sigma \).
  - Definition, next slide.
- \( U = Y(\Sigma^+)^2X^T \).
If $\Sigma$ is a diagonal matrix, its Moore-Penrose inverse is another diagonal matrix whose $i$-th entry is:

- $1/\sigma$ if $\sigma$ is not 0.
- 0 if $\sigma$ is 0.

**Example:**

$\Sigma = \begin{bmatrix} 4 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 0 \end{bmatrix}$

$\Sigma^+ = \begin{bmatrix} 0.25 & 0 & 0 \\ 0 & 0.5 & 0 \\ 0 & 0 & 0 \end{bmatrix}$
To decrease the expected error between $M$ and its decomposition, we must pick rows and columns in a nonuniform manner.

The *importance* of a row or column of $M$ is the square of its Frobenius norm.

- That is, the sum of the squares of its elements.

When picking rows and columns, the probabilities must be proportional to importance.

**Example**: $[3, 4, 5]$ has importance 50, and $[3, 0, 1]$ has importance 10, so pick the first 5 times as often as the second.