Data Mining and Matrices
03 – Singular Value Decomposition

Rainer Gemulla, Pauli Miettinen

April 25, 2013
The SVD is the Swiss Army knife of matrix decompositions

—Diane O’Leary, 2006
Outline

1 The Definition

2 Properties of the SVD

3 Interpreting SVD

4 SVD and Data Analysis
   - How many factors?
   - Using SVD: Data processing and visualization

5 Computing the SVD

6 Wrap-Up

7 About the assignments
The definition

**Theorem.** For every $A \in \mathbb{R}^{m \times n}$ there exists $m \times m$ orthogonal matrix $U$ and $n \times n$ orthogonal matrix $V$ such that $U^TAV$ is an $m \times n$ diagonal matrix $\Sigma$ that has values $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_{\min\{n,m\}} \geq 0$ in its diagonal.

- I.e. every $A$ has decomposition $A = U\Sigma V^T$
  - The **singular value decomposition** (SVD)
- The values $\sigma_i$ are the **singular values** of $A$
- Columns of $U$ are the **left singular vectors** and columns of $V$ the **right singular vectors** of $A$
Outline

1. The Definition
2. Properties of the SVD
3. Interpreting SVD
4. SVD and Data Analysis
   - How many factors?
   - Using SVD: Data processing and visualization
5. Computing the SVD
6. Wrap-Up
7. About the assignments
The fundamental theorem of linear algebra

The fundamental theorem of linear algebra states that every matrix $A \in \mathbb{R}^{m \times n}$ induces four fundamental subspaces:

- **The range** of dimension $\text{rank}(A) = r$
  - The set of all possible linear combinations of columns of $A$

- **The kernel** of dimension $n - r$
  - The set of all vectors $x \in \mathbb{R}^n$ for which $Ax = 0$

- **The coimage** of dimension $r$

- **The cokernel** of dimension $m - r$

The bases for these subspaces can be obtained from the SVD:

- Range: the first $r$ columns of $U$
- Kernel: the last $(n - r)$ columns of $V$
- Coimage: the first $r$ columns of $V$
- Cokernel: the last $(m - r)$ columns of $U$
Pseudo-inverses

Problem.
Given $A \in \mathbb{R}^{m \times n}$ and $b \in \mathbb{R}^m$, find $x \in \mathbb{R}^n$ minimizing $\|Ax - b\|_2$.

- If $A$ is invertible, the solution is $A^{-1}Ax = A^{-1}b \iff x = A^{-1}b$
- A **pseudo-inverse** $A^+$ captures some properties of the inverse $A^{-1}$
- The **Moose–Penrose pseudo-inverse** of $A$ is a matrix $A^+$ satisfying the following criteria
  - $AA^+A = A$ (but it is possible that $AA^+ \neq I$)
  - $A^+AA^+ = A^+$ (cf. above)
  - $(AA^+)^T = AA^T$ ($AA^+$ is symmetric)
  - $(A^+A)^T = A^+A$ (as is $A^+A$)

- If $A = U\Sigma V^T$ is the SVD of $A$, then $A^+ = V\Sigma^{-1}U^T$
  - $\Sigma^{-1}$ replaces $\sigma_i$'s with $1/\sigma_i$ and transposes the result

Theorem.
The optimum solution for the above problem can be obtained using $x = A^+b$. 

Truncated (thin) SVD

- The rank of the matrix is the number of its non-zero singular values
  - Easy to see by writing $A = \sum_{i=1}^{\min\{n,m\}} \sigma_i u_i v_i^T$

- The truncated (or thin) SVD only takes the first $k$ columns of $U$ and $V$ and the main $k \times k$ submatrix of $\Sigma$
  - $A_k = \sum_{i=1}^{k} \sigma_i u_i v_i^T = U_k \Sigma_k V_k^T$
  - $\text{rank}(A_k) = k$ (if $\sigma_k > 0$)
  - $U_k$ and $V_k$ are no more orthogonal, but they are column-orthogonal

- The truncated SVD gives a low-rank approximation of $A$

\[ A \approx U \Sigma V^T \]
SVD and matrix norms

Let \( A = U \Sigma V^T \) be the SVD of \( A \). Then

- \( \| A \|_F^2 = \sum_{i=1}^{\min\{n,m\}} \sigma_i^2 \)
- \( \| A \|_2 = \sigma_1 \)
  - Remember: \( \sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_{\min\{n,m\}} \geq 0 \)
- Therefore \( \| A \|_2 \leq \| A \|_F \leq \sqrt{n} \| A \|_2 \)
- The Frobenius of the truncated SVD is \( \| A_k \|_F^2 = \sum_{i=1}^{k} \sigma_i^2 \)
  - And the Frobenius of the difference is \( \| A - A_k \|_F^2 = \sum_{i=k+1}^{\min\{n,m\}} \sigma_i^2 \)

The Eckart–Young theorem

Let \( A_k \) be the rank-\( k \) truncated SVD of \( A \). Then \( A_k \) is the closest rank-\( k \) matrix of \( A \) in the Frobenius sense. That is

\[
\| A - A_k \|_F \leq \| A - B \|_F \quad \text{for all rank-}k \text{ matrices } B.
\]
Eigendecompositions

- An **eigenvector** of a square matrix \( A \) is a vector \( v \) such that \( A \) only changes the magnitude of \( v \)
  - I.e. \( Av = \lambda v \) for some \( \lambda \in \mathbb{R} \)
  - Such \( \lambda \) is an **eigenvalue** of \( A \)

- The **eigendecomposition** of \( A \) is \( A = Q \Delta Q^{-1} \)
  - The columns of \( Q \) are the eigenvectors of \( A \)
  - Matrix \( \Delta \) is a diagonal matrix with the eigenvalues

- Not every (square) matrix has eigendecomposition
  - If \( A \) is of form \( BB^T \), it always has eigendecomposition

- The SVD of \( A \) is closely related to the eigendecompositions of \( AA^T \) and \( A^T A \)
  - The left singular vectors are the eigenvectors of \( AA^T \)
  - The right singular vectors are the eigenvectors of \( A^T A \)
  - The singular values are the square roots of the eigenvalues of both \( AA^T \) and \( A^T A \)
Outline

1. The Definition
2. Properties of the SVD
3. Interpreting SVD
4. SVD and Data Analysis
   - How many factors?
   - Using SVD: Data processing and visualization
5. Computing the SVD
6. Wrap-Up
7. About the assignments
Factor interpretation

- The most common way to interpret SVD is to consider the columns of $U$ (or $V$)
  - Let $A$ be objects-by-attributes and $U\Sigma V^T$ its SVD
  - If two columns have similar values in a row of $V^T$, these attributes are somehow similar (have strong correlation)
  - If two rows have similar values in a column of $U$, these users are somehow similar

![Figure 3.2. The first two factors for a dataset ranking wines.](image)

- Example: people’s ratings of different wines
- Scatterplot of first and second column of $U$
  - left: likes wine
  - right: doesn’t like
  - up: likes red wine
  - bottom: likes white vine

Conclusion: winelovers like red and white, others care more
Geometric interpretation

- Let $U \Sigma V^T$ be the SVD of $M$
- SVD shows that every linear mapping $y = Mx$ can be considered as a series of rotation, stretching, and rotation operations
  - Matrix $V^T$ performs the first rotation $y_1 = V^T x$
  - Matrix $\Sigma$ performs the stretching $y_2 = \Sigma y_1$
  - Matrix $U$ performs the second rotation $y = Uy_2$

$M = U \cdot \Sigma \cdot V^*$
Dimension of largest variance

- The singular vectors give the dimensions of the variance in the data
  - The first singular vector is the dimension of the largest variance
  - The second singular vector is the orthogonal dimension of the second largest variance
    - First two dimensions span a hyperplane

- From Eckart–Young we know that if we project the data to the spanned hyperplanes, the distance of the projection is minimized
Component interpretation

- Recall that we can write $A = U\Sigma V^T = \sum_{i=1}^{r'} \sigma_i u_i v_i^T = \sum_{i=1}^{r'} A_i$
  - $A_i = \sigma_i v_i u_i^T$

- This explains the data as a sums of (rank-1) layers
  - The first layer explains the most
  - The second corrects that by adding and removing smaller values
  - The third corrects that by adding and removing even smaller values
  - ...

- The layers don’t have to be very intuitive
Outline

1. The Definition
2. Properties of the SVD
3. Interpreting SVD
4. SVD and Data Analysis
   - How many factors?
   - Using SVD: Data processing and visualization
5. Computing the SVD
6. Wrap-Up
7. About the assignments
Outline

1. The Definition
2. Properties of the SVD
3. Interpreting SVD
4. SVD and Data Analysis
   - How many factors?
   - Using SVD: Data processing and visualization
5. Computing the SVD
6. Wrap-Up
7. About the assignments
Problem

- Most data mining applications do not use full SVD, but truncated SVD
  - To concentrate on “the most important parts”
- But how to select the rank $k$ of the truncated SVD?
  - What is important, what is unimportant?
  - What is structure, what is noise?
  - Too small rank: all subtlety is lost
  - Too big rank: all smoothing is lost
- Typical methods rely on singular values in a way or another
Perhaps the oldest method is the Guttman–Kaiser criterion:
  ▶ Select $k$ so that for all $i > k$, $\sigma_i < 1$
  ▶ Motivation: all components with singular value less than unit are uninteresting

Another common method is to select enough singular values such that the sum of their squares is 90% of the total sum of the squared singular values
  ▶ The exact percentage can be different (80%, 95%)
  ▶ Motivation: The resulting matrix “explains” 90% of the Frobenius norm of the matrix (a.k.a. energy)

**Problem:** Both of these methods are based on arbitrary thresholds and do not consider the “shape” of the data
Cattell’s Scree test

- The **scree plot** plots the singular values in decreasing order
  - The plot looks like a side of the hill, thence the name
- The scree test is a subjective decision on the rank based on the shape of the scree plot
- The rank should be set to a point where
  - there is a clear drop in the magnitudes of the singular values; or
  - the singular values start to even out
- **Problem:** Scree test is subjective, and many data don’t have any clear shapes to use (or have many)
  - Automated methods have been developed to detect the shapes from the scree plot
Entropy-based method

- Consider the relative contribution of each singular value to the overall Frobenius norm
  - Relative contribution of $\sigma_k$ is $f_k = \sigma_k^2 / \sum_i \sigma_i^2$
- We can consider these as probabilities and define the (normalized) entropy of the singular values as

$$E = -\frac{1}{\log(\min\{n, m\})} \sum_{i=1}^{\min\{n, m\}} f_i \log f_i$$

- The basis of the logarithm doesn’t matter
- We assume that $0 \cdot \infty = 0$
- Low entropy (close to 0): the first singular value has almost all mass
- High entropy (close to 1): the singular values are almost equal

- The rank is selected to be the smallest $k$ such that $\sum_{i=1}^k f_i \geq E$
- **Problem:** Why entropy?
Random flip of signs

- Multiply every element of the data $\mathbf{A}$ randomly with either 1 or $-1$ to get $\tilde{\mathbf{A}}$
  - The Frobenius norm doesn’t change ($\|\mathbf{A}\|_F = \|\tilde{\mathbf{A}}\|_F$)
  - The spectral norm does change ($\|\mathbf{A}\|_2 \neq \|\tilde{\mathbf{A}}\|_2$)
    - How much this changes depends on how much “structure” $\mathbf{A}$ has
- We try to select $k$ such that the residual matrix contains only noise
  - The residual matrix contains the last $m - k$ columns of $\mathbf{U}$, $\min\{n, m\} - k$ singular values, and last $n - k$ rows of $\mathbf{V}^T$
  - If $\mathbf{A}_{-k}$ is the residual matrix of $\mathbf{A}$ after rank-$k$ truncated SVD and $\tilde{\mathbf{A}}_{-k}$ is that for the matrix with randomly flipped signs, we select rank $k$ to be such that $\frac{\|\mathbf{A}_{-k}\|_2 - \|\tilde{\mathbf{A}}_{-k}\|_2}{\|\mathbf{A}_{-k}\|_F}$ is small
- **Problem:** How small is small?
Outline

1. The Definition
2. Properties of the SVD
3. Interpreting SVD
4. SVD and Data Analysis
   - How many factors?
   - Using SVD: Data processing and visualization
5. Computing the SVD
6. Wrap-Up
7. About the assignments
Normalization

- Data should usually be normalized before SVD is applied
  - If one attribute is height in meters and other weights in grams, weight seems to carry much more importance in data about humans
  - If data is all positive, the first singular vector just explains where in the positive quadrant the data is
- The $z$-scores are attributes whose values are transformed by
  - centering them to 0
    - Remove the mean of the attribute’s values from each value
  - normalizing the magnitudes
    - Divide every value with the standard deviation of the attribute
- Notice that the $z$-scores assume that
  - all attributes are equally important
  - attribute values are approximately normally distributed
- Values that have larger magnitude than importance can also be normalized by first taking logarithms (from positive values) or cubic roots
- The effects of normalization should always be considered
Removing noise

- Very common application of SVD is to remove the noise from the data
- This works simply by taking the truncated SVD from the (normalized) data
  - The big problem is to select the rank of the truncated SVD
- Example:

  ![Original data](image1)
  - Original data
    - Looks like 1-dimensional with some noise
  - The right singular vectors show the directions
    - The first looks like the data direction
    - The second looks like the noise direction
  - The singular values confirm this

  \[
  \sigma_1 = 11.73 \\
  \sigma_2 = 1.71
  \]
Removing dimensions

- Truncated SVD can also be used to battle the curse of dimensionality
  - All points are close to each other in very high dimensional spaces
  - High dimensionality slows down the algorithms
- Typical approach is to work in a space spanned by the columns of $V^T$
  - If $U \Sigma V^T$ is the SVD of $A \in \mathbb{R}^{m \times n}$, project $A$ to $AV_k \in \mathbb{R}^{m \times k}$ where $V_k$ has the first $k$ columns of $V$
  - This is known as the Karhunen–Loève transform (KLT) of the rows of $A$
    - Matrix $A$ must be normalized to z-scores in KLT
Visualization

- Truncated SVD with \( k = 2, 3 \) allows us to visualize the data
  - We can plot the projected data points after 2D or 3D Karhunen–Loève transform
  - Or we can plot the scatter plot of two or three (first, left/right) singular vectors

Figure 3.2. The first two factors for a dataset ranking wines.
Latent semantic analysis

- The **latent semantic analysis** (LSA) is an information retrieval method that uses SVD.
- The data: a term–document matrix $A$
  - the values are (weighted) term frequencies
  - typically tf/idf values (the frequency of the term in the document divided by the global frequency of the term)
- The truncated SVD $A_k = U_k \Sigma_k V_k^T$ of $A$ is computed
  - Matrix $U_k$ associates documents to topics
  - Matrix $V_k$ associates topics to terms
  - If two rows of $U_k$ are similar, the corresponding documents “talk about same things”
- A query $q$ can be answered by considering its term vector $q$
  - $q$ is projected to $q_k = qV\Sigma^{-1}$
  - $q_k$ is compared to rows of $U$ and most similar rows are returned
Outline

1. The Definition
2. Properties of the SVD
3. Interpreting SVD
4. SVD and Data Analysis
   - How many factors?
   - Using SVD: Data processing and visualization
5. Computing the SVD
6. Wrap-Up
7. About the assignments
Algorithms for SVD

- In principle, the SVD of $\mathbf{A}$ can be computed by computing the eigendecomposition of $\mathbf{AA}^T$
  - This gives us left singular vectors and squares of singular values
  - Right singular vectors can be solved: $\mathbf{V}^T = \mathbf{\Sigma}^{-1}\mathbf{U}^T\mathbf{A}$
  - **Bad for numerical stability!**
- Full SVD can be computed in time $O(nm \min\{n, m\})$
  - Matrix $\mathbf{A}$ is first reduced to a bidiagonal matrix
  - The SVD of the bidiagonal matrix is computed using iterative methods (similar to eigendecompositions)
- Methods that are faster in practice exist
  - Especially for truncated SVD
- Efficient implementation of an SVD algorithm requires considerable work and knowledge
  - Luckily (almost) all numerical computation packages and programs implement SVD
Lessons learned

- SVD is the Swiss Army knife of (numerical) linear algebra
  - ranks, kernels, norms, . . .

- SVD is also very useful in data analysis
  - noise removal, visualization, dimensionality reduction, . . .

- Selecting the correct rank for truncated SVD is still a problem
Suggested reading

- Skillicorn, Ch. 3
  - Excellent source for the algorithms and theory, but very dense
Basic information

- Assignment sheet will be made available later today/early tomorrow
  - We’ll announce it in the mailing list
- DL in two weeks, delivery by e-mail
  - Details in the assignment sheet
- Hands-on assignment: data analysis using SVD
- Recommended software: R
  - Good alternatives: Matlab (commercial), GNU Octave (open source), and Python with NumPy, SciPy, and matplotlib (open source)
  - Excel is not a good alternative (too complicated)
- What you have to return?
  - Single document that answers to all questions (all figures, all analysis of the results, the main commands you used for the analysis if asked)
  - Supplementary material containing the transcript of all commands you issued/all source code
  - Both files in PDF format