Singular Value Decomposition:
Theory and Applications

Daniel Khashabi

Spring 2015
Last Update: March 2, 2015

1 Introduction

$$A = UDV^\top$$

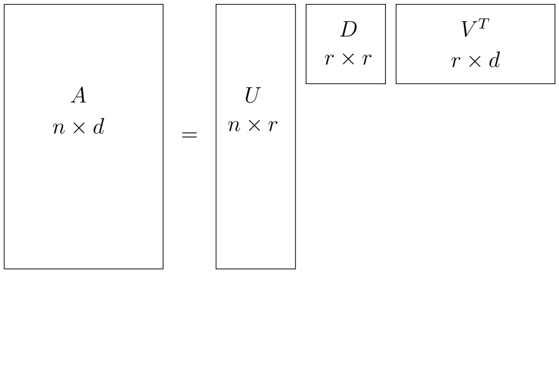
where columns of U and V are orthonormal and matrix D is diagonal with positive real values. The diagonal elements of D are called singular values. The m rows of U are called left-singular vectors and d rows of V are called right-singular vectors.

The SVD of A gives the best rank k approximation to A with respect to squared-norm, for any k.

Remark 1. *SVD is defined for all matrices, whereas the more commonly used Eigenvector Decomposition requires the matrix A be square and certain other conditions on the matrix to ensure orthogonality of the eigenvectors.*

- The left-singular vectors of A are eigenvectors of $AA^\top$.
- The right-singular vectors of A are eigenvectors of $A^\top A$.
- The non-zero singular values of A (found on the diagonal entries of D) are the square roots of the non-zero eigenvalues of both $A^\top A$ and $AA^\top$.

Lemma 1. *Suppose $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_r$ are left singular vectors as a result of SVD. It can be shown that these vectors satisfy the following maximizations:*



$$\mathbf{v}_1 = \arg\max_{|\mathbf{v}|=1} |A\mathbf{v}|$$

$$\mathbf{v}_2 = \arg\max_{\substack{|\mathbf{v}|=1\\ \mathbf{v}\perp\mathbf{v}_1}} |A\mathbf{v}|$$

$$\mathbf{v}_3 = \arg\max_{\substack{|\mathbf{v}|=1\\ \mathbf{v}\perp\mathbf{v}_1\\ \mathbf{v}\perp\mathbf{v}_2}} |A\mathbf{v}|$$

$$\vdots$$

$$\mathbf{v}_r = \arg\max_{\substack{|\mathbf{v}|=1\\ \mathbf{v}\perp\mathbf{v}_1\\ \dots\\ \mathbf{v}\perp\mathbf{v}_{r-1}}} |A\mathbf{v}|$$

It can be shown that $\sigma_1(A) = |A\mathbf{v}_1|$ is the diagonal element of D (or the first singular value). Similarly for other singular values. The lemma gets translated into the following Theorem.

Theorem 1. *For $1 \leq k \leq r$, let V_k be the subspace spanned by $\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_k$. For each k, V_k is the best-fit k-dimensional subspace for matrix A.*

Lemma 2. *For any matrix A, the sum of squares of the singular values equals the square of the Frobenius norm. That is,*

$$\sum_i {\sigma_i}^2(A) = \sum_{i,j} a_{i,j}^2 = \|A\|_F^2$$

Lemma 3. *It can be shown the following relation between the singular values and right/left-singular vectors:*

$$\mathbf{u}_i = \frac{1}{\sigma_i(A)} A\mathbf{v}_i$$

Theorem 2. *The left-singular vectors are pairwise orthogonal.*

Theorem 3. *For any matrix B of rank at most k:*

$$\|A - A_k\|_F \leq \|A - B\|_F$$

Lemma 4.

$$\|A - A_k\|_2^2 = \sigma_{k+1}^2$$

Example 1 (Interpreting an SVD on reviews' matrix)**.** *Suppose matrix A is matrix of costumer-restaurant ratings (rows being persons and columns being restaurants).*
Left singular vectors, $\mathbf{u}_i$ have size "the number of people", and can be seen as the orthogonal directions of reviews by people. Specifically, the first left singular vector, corresponds to the most popular direction/pattern of reviews by people.
Similarly, right singular vectors, $\mathbf{v}_i$ have size "the number of restaurants", and can be seen as the orthogonal directions of reviews given to restaurants. For example, the first right singular vector, corresponds to the most popular direction of reviews to restaurants.

The singular values show the popularity of directions/review patterns. If $\sigma_1 \gg \sigma_2$ it shows that there is a consensus in the reviews by people for restaurants. If $\sigma_1 \approx \sigma_2 \gg \sigma_3$, it shows that there is two major scoring patterns. The bigger the singular gap $\sigma_1 - \sigma_2$ is, the more consensus exists in the reviews.
The definition of "consensus" here is delicate. The consensus here is defined based on the "directionality" of the reviews. In other words, SVD does not care whether the reviews are big or small (or positive or negative). Instead it values the consensus in terms how coherent the reviews are in one specific direction (in customer-restaurant space).

2 Power Method

Consider an arbitrary matrix B. The power iteration algorithm starts with a vector $\mathbf{x}_0$, and gives an approximation to the dominant eigenvector, if converges at all (otherwise a random vector). Given the initialization $\mathbf{x}_0$, the updates of the algorithm are the followings:

$$\mathbf{x}_{k+1} = \frac{B\mathbf{x}_k}{\|B\mathbf{x}_k\|}$$

The convergence is guaranteed under the following two conditions:

- B has an eigenvalue strictly greater in magnitude than its other eigenvalues.
- The starting vector $\mathbf{x}_0$ has a nonzero component in the direction of an dominant eigenvector.

The next lemma explains how the Power Iterations is useful for SVD.

Lemma 5. *Given a matrix $B = AA^\top$, the power iteration algorithm on converges to $\mathbf{u}_1$, the first singular vector of A, if converges at all.*

Proof. Consider an SVD of A:

$$A = \sum_i \sigma_i \mathbf{u}_i \mathbf{v}_i^\top$$

Then for B we have:

$$B = AA^\top = \left(\sum_i \sigma_i \mathbf{u}_i \mathbf{v}_i^\top\right)\left(\sum_i \sigma_i \mathbf{u}_i \mathbf{v}_i^\top\right)^\top = \sum_{i,j} \sigma_i \sigma_j \mathbf{u}_i \mathbf{u}_i^\top$$

Repeating the multiplication:

$$B^2 = \left(\sum_{i,j} \sigma_i \sigma_j \mathbf{u}_i \mathbf{u}_i^\top\right)\left(\sum_{i,j} \sigma_i \sigma_j \mathbf{u}_i \mathbf{u}_i^\top\right) = \sum_{i,j} \sigma_i^2 \sigma_j^2 \mathbf{u}_i \left(\mathbf{u}_i^\top \mathbf{u}_j\right) \mathbf{u}_j^\top = \sum_i \sigma_i^4 \mathbf{u}_i \mathbf{u}_i^\top$$

Since $\mathbf{u}_i^\top \mathbf{u}_j = 0$, for $i \neq j$. Similarly we have:

$$B^k = \sum_i \sigma_i^{2i} \mathbf{u}_i \mathbf{u}_i^\top$$

If $\sigma_1 > \sigma_2$, as k grows we have the following convergence:

$$B^k \to \sigma_1^{2k} \mathbf{u}_1 \mathbf{u}_1^\top$$

By proper normalization of the rank-1 matrix $\mathbf{u}_1 \mathbf{u}_1^\top$ and a little algebra, we can find $\mathbf{u}_1$. However the issue with the above method is that we need to handle the matrix $\mathbf{u}_1 \mathbf{u}_1^\top$ which can be very large in practice. The trick is that, instead of computing

$$B^k \to \sigma_1^{2k} \mathbf{u}_1 \mathbf{u}_1^\top$$

we choose a random point $\mathbf{x}_0 = \sum_i \alpha_i \mathbf{u}_i$ and calculate $B^k \mathbf{x}_0$.

$$B^k \mathbf{x}_0 \to \sigma_1^{2k} \mathbf{u}_1 \mathbf{u}_1^\top \left(\sum_i \alpha_i \mathbf{u}_i \right) = \alpha_1 \sigma_1^{2k} \mathbf{u}_1$$

Which gives $\mathbf{u}_1$, after a normalization over a vector. Note that starting from a random vector calculation of powers need a “matrix × vector” operation, which computationally is a moderate operation. □

Example 2. *We show a sample run of power iteration. Suppose:*

$$A = \begin{bmatrix} +1 & +2 \\ -1 & +2 \\ +1 & -2 \\ -1 & -2 \end{bmatrix}$$

and we are starting from $x = [11]^\top$ and $k = 3$. We repeat the power method for three steps:

1. $x_0 = [1, 1]^\top$
2. *for* $k = \{1, 2, 3\}$
3. $\quad x_k = Ax_{k-1} / \|Ax_{k-1}\|$

The results are:

$$x_0 = [1, 1]^\top \to x_1 = [0.24, 0.97]^\top \to x_2 = [0.06, 0.99]^\top \to x_3 = [0.01, 0.99]^\top$$

We will see that the direct calculations will give $\mathbf{u} = [0, 1]^\top$ which is very close to x_3.

We can computing the exact values directly:

$$B = A^\top A = \begin{bmatrix} 4 & 0 \\ 0 & 16 \end{bmatrix}$$

We find the solutions to $B\mathbf{v} = \lambda \mathbf{v}$. The eigenvalues and eigenvectors of B are:

$$[\mathbf{u}_1, \mathbf{u}_2] = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$[\lambda_1, \lambda_2] = [16, 4]$$

The eigenvectors of $A^\top A$ are the right singular vectors of A. Also the singular values equal to the eigenvalues. Since $\mathbf{v}_i = \frac{1}{\sigma_i(A)} A\mathbf{u}_i$, the two other left singular vectors are:

$$[\mathbf{v}_1, \mathbf{v}_2] = \begin{bmatrix} -0.5 & -0.5 \\ -0.5 & +0.5 \\ +0.5 & -0.5 \\ +0.5 & +0.5 \end{bmatrix}$$

Lemma 6. *For any given A, $\mathbf{u}_1$ is the first singular vector of $A^\top$ (suppose $\sigma_1 > \sigma_2$)*

Proof. □

2.1 Further topics

2.1.1 Missing data in SVD

Consider the matrix of reviews in example 1. If some people refuse to give reviews for some restaurants, we will have some missing values in our matrix. The question is, how to handle the missing values when doing SVD.

In many practical applications, the missing values are replaced with zeros. But still there seems to be a need for methods which give grantees on the results, while being practical and fast.

One other trick is using regularizer (say an l_2 regularizer) in the objective function. In other words, instead of using SVD directly which minimizes the Frobenius norm type object, we can augment the objective with a regularizer and minimize it directly (say with gradient descent). In this description, this corresponds to minimizing the components of the norm which can be measured, i.e. those which have known values. The regularization term can be seen as a Bayesian prior on the components of the feature vectors, with the SVD calculating the maximum likelihood estimator, subject to this prior and the known values.

3 Bibliographical notes