Singular Value Decomposition

(Com S 477/577 Notes)

Yan-Bin Jia

Sep 12, 2017

1 Introduction

Now comes a highlight of linear algebra. Any real $m \times n$ matrix can be factored as

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

where U is an $m \times m$ orthogonal matrix[1] whose columns are the eigenvectors of AA^T, V is an $n \times n$ orthogonal matrix whose columns are the eigenvectors of A^TA, and Σ is an $m \times n$ diagonal matrix of the form

$$\Sigma = \begin{pmatrix} \sigma_1 & & & & \\ & \ddots & & 0 & \\ & & \sigma_r & & \\ & & & 0 & \\ & 0 & & & \ddots \\ & & & & & 0 \end{pmatrix}$$

with $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r > 0$ and $r = \text{rank}(A)$. In the above, $\sigma_1, \ldots, \sigma_r$ are the square roots of the eigenvalues of A^TA. They are called the *singular values* of A.

Our basic goal is to "solve" the system $A\boldsymbol{x} = \boldsymbol{b}$ for all matrices A and vectors $\boldsymbol{b}$. A second goal is to solve the system using a numerically stable algorithm. A third goal is to solve the system in a reasonably efficient manner. For instance, we do not want to compute A^{-1} using determinants.

Three situations arise regarding the basic goal:

(a) If A is square and invertible, we want to have the solution $\boldsymbol{x} = A^{-1}\boldsymbol{b}$.

(b) If A is underconstrained, we want the entire set of solutions.

(c) If A is overconstrained, we could simply give up. But this case arises a lot in practice, so instead we will ask for the least-squares solution. In other words, we want that $\bar{\boldsymbol{x}}$ which minimizes the error $\|A\bar{\boldsymbol{x}} - \boldsymbol{b}\|$. Geometrically, $A\bar{\boldsymbol{x}}$ is the point in the column space of A closest to $\boldsymbol{b}$.

[1] That is, $UU^T = U^TU = I$.

Gaussian elimination is reasonably efficient, but it is not numerically very stable. In particular, elimination does not deal with nearly singular matrices. The method is not designed for overconstrained systems. Even for underconstrained systems, the method requires extra work.

The poor numerical character of elimination can be seen in a couple ways. First, the elimination process assumes a non-singular matrix. But singularity, and rank in general, is a slippery concept. After all, the matrix A contains continuous, possibly noisy, entries. Yet, rank is a discrete integer. Strictly speaking, the two sets below are linearly independent vectors:

$$\left\{ \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \right\}, \qquad \left\{ \begin{pmatrix} 1.01 \\ 1.00 \\ 1.00 \end{pmatrix}, \begin{pmatrix} 1.00 \\ 1.01 \\ 1.00 \end{pmatrix}, \begin{pmatrix} 1.00 \\ 1.00 \\ 1.01 \end{pmatrix} \right\}.$$

Yet, the first set seems genuinely independent, while the second set seems "almost dependent".

Second, elimination-based methods work like LU decomposition, which represents the coefficient matrix A as a matrix product LDU, where L and U are respectively lower and upper diagonal and D is diagonal. One solves the system $A\boldsymbol{x} = \boldsymbol{b}$ by solving (via backsubstitution) $L\boldsymbol{y} = \boldsymbol{b}$ and $U\boldsymbol{x} = D^{-1}\boldsymbol{y}$. If A is nearly singular, then D will contain near-zero entries on its diagonal, and thus D^{-1} will contain large numbers. That is OK since in principle one needs the large numbers to obtain a true solution. The problem is if A contains noisy entries. Then the large numbers may be pure noise that dominates the true information. Furthermore, since L and U can be fairly arbitrary, they may distort or magnify that noise across other variables.

Singular value decomposition is a powerful technique for dealing with sets of equations or matrices that are either singular or else numerically very close to singular. In many cases where Gaussian elimination and LU decomposition fail to give satisfactory results, SVD will not only diagnose the problem but also give you a useful numerical answer. It is also the method of choice for solving most linear least-squares problems.

2 SVD Close-up

An $n \times n$ symmetric matrix A has an eigen decomposition in the form of

$$A = S\Lambda S^{-1},$$

where Λ is a diagonal matrix with the eigenvalues δ_i of A on the diagonal and S contains the eigenvectors of A.

Why is the above decomposition appealing? The answer lies in the change of coordinates $\boldsymbol{y} = S^{-1}\boldsymbol{x}$. Instead of working with the system $A\boldsymbol{x} = \boldsymbol{b}$, we can work with the system $\Lambda\boldsymbol{y} = \boldsymbol{c}$, where $\boldsymbol{c} = S^{-1}\boldsymbol{b}$. Since Λ is diagonal, we are left with a trivial system

$$\delta_i y_i = c_i, \qquad i = 1, \ldots, n.$$

If this system has a solution, then another change of coordinates gives us $\boldsymbol{x}$, that is, $\boldsymbol{x} = S\boldsymbol{y}$.[2]

[2]There is no reason to believe that computing $S^{-1}\boldsymbol{b}$ is any easier than computing $A^{-1}\boldsymbol{b}$. However, in the ideal case, the eigenvectors of A are orthogonal. This is true, for instance, if $AA^T = A^TA$. In that case the columns of S are orthogonal and so we can take S to be orthogonal. But then $S^{-1} = S^T$, and the problem of solving $A\boldsymbol{x} = \boldsymbol{b}$ becomes very simple.

Unfortunately, the decomposition $A = S\Lambda S^{-1}$ is not always possible. The condition for its existence is that A is $n \times n$ symmetric with n linearly independent eigenvectors. Even worse, what do we do if A is not square?

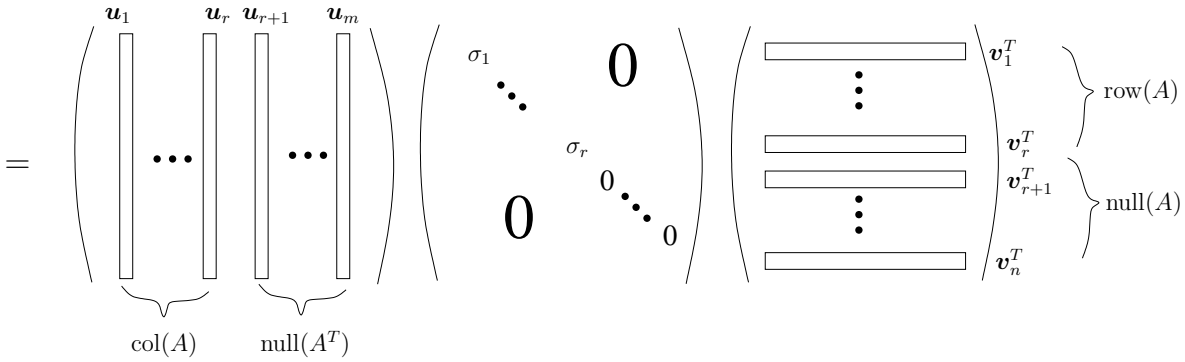
The answer is work with A^TA and AA^T, both of which are symmetric (and have n and m orthogonal eigenvectors, respectively). So we have the following decompositions:

$$\begin{aligned} A^TA &= VDV^T, \\ AA^T &= UD'U^T, \end{aligned}$$

where V is an $n \times n$ orthogonal matrix consisting of the eigenvectors of A^TA, D an $n \times n$ diagonal matrix with the eigenvalues of A^TA on the diagonal, U an $m \times m$ orthogonal matrix consisting of the eigenvectors of AA^T, and D' an $m \times m$ diagonal matrix with the eigenvalues of AA^T on the diagonal. It turns out that D and D' have the same non-zero diagonal entries except that the order might be different.

Recall the SVD form of A:

$$\underset{m \times n}{A} = \underset{m \times m}{U} \quad \underset{m \times n}{\Sigma} \quad \underset{n \times n}{V^T} \tag{1}$$



There are several facts about SVD:

(a) $\text{rank}(A) = \text{rank}(\Sigma) = r$.

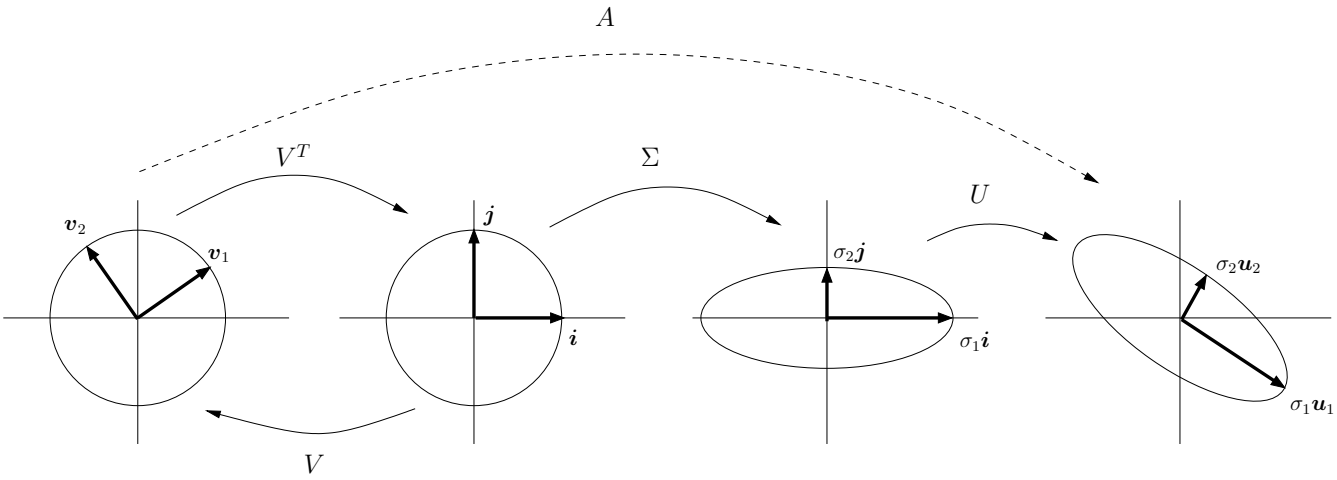
(b) The column space of A is spanned by the first r columns of U.

(c) The null space of A is spanned by the last $n - r$ columns of V.

(d) The row space of A is spanned by the first r columns of V.

(e) The null space of A^T is spanned by the last $m - r$ columns of U.

We can think of U and V as rotations and reflections and Σ as stretching matrix. The next figure illustrates the sequence of transformation under A on the unit vectors $\boldsymbol{v}_1$ and $\boldsymbol{v}_2$ (and all other vectors on the unit circle) in the case $m = n = 2$. Note that $V = (\boldsymbol{v}_1\boldsymbol{v}_2)$. When multiplied by V^T on the left, the two vectors undergo a rotation and become unit vectors $\boldsymbol{i} = (1, 0)^T$ and $\boldsymbol{j} = (0, 1)^T$. Then the matrix Σ stretches these two resulting vectors to $\sigma_1\boldsymbol{i}$ and $\sigma_2\boldsymbol{j}$, respectively. In the last step, the vectors undergo a final rotation due to U and become $\sigma_1\boldsymbol{u}_1$ and $\sigma_2\boldsymbol{u}_2$.



From (1) we also see that

$$A = \sigma_1 \boldsymbol{u}_1 \boldsymbol{v}_1^T + \cdots + \sigma_r \boldsymbol{u}_r \boldsymbol{v}_r^T.$$

We can swap σ_i with σ_j as long as we swap $\boldsymbol{u}_i$ with $\boldsymbol{u}_j$ and $\boldsymbol{v}_i$ with $\boldsymbol{v}_j$ at the same time. If $\sigma_i = \sigma_j$, then $\boldsymbol{u}_i$ and $\boldsymbol{u}_j$ can be swapped as long as $\boldsymbol{v}_i$ and $\boldsymbol{v}_j$ are also swapped. SVD is unique up to the permutations of $(\boldsymbol{u}_i, \sigma_i, \boldsymbol{v}_i)$ or of $(\boldsymbol{u}_i, \boldsymbol{v}_i)$ among those with equal σ_is. It is also unique up to the signs of $\boldsymbol{u}_i$ and $\boldsymbol{v}_i$, which have to change simultaneously.

It follows that

$$\begin{aligned} A^T A &= V \Sigma^T U^T U \Sigma V^T \\ &= V \begin{pmatrix} \sigma_1^2 & & & & & \\ & \ddots & & & & \\ & & \sigma_r^2 & & & \\ & & & 0 & & \\ & & & & \ddots & \\ & & & & & 0 \end{pmatrix} V^T. \end{aligned}$$

Hence $\sigma_1^2, \ldots, \sigma_r^2$ (and 0 if $r < n$) are the eigenvalues of $A^T A$, which is positive definite if $\operatorname{rank}(A) = n$, and $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_n$ its eigenvectors.

Similarly, we have

$$AA^T = U \begin{pmatrix} \sigma_1^2 & & & & & \\ & \ddots & & & & \\ & & \sigma_r^2 & & & \\ & & & 0 & & \\ & & & & \ddots & \\ & & & & & 0 \end{pmatrix} U^T.$$

Therefore $\sigma_1^2, \ldots, \sigma_r^2$ (and 0 if $r < m$) are also eigenvalues of AA^T, and $\boldsymbol{u}_1, \ldots, \boldsymbol{u}_m$ its eigenvectors.

EXAMPLE 1. Find the singular value decomposition of

$$A = \begin{pmatrix} 2 & 2 \\ -1 & 1 \end{pmatrix}.$$

The eigenvalues of

$$A^T A = \begin{pmatrix} 5 & 3 \\ 3 & 5 \end{pmatrix}$$

are 2 and 8 corresponding to unit eigenvectors

$$\boldsymbol{v}_1 = \begin{pmatrix} -\frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{pmatrix} \qquad \text{and} \qquad \boldsymbol{v}_2 = \begin{pmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{pmatrix},$$

respectively. Hence $\sigma_1 = \sqrt{2}$ and $\sigma_2 = \sqrt{8} = 2\sqrt{2}$. We have

$$\begin{aligned} A\boldsymbol{v}_1 &= \sigma_1 \boldsymbol{u}_1 \boldsymbol{v}_1^T \boldsymbol{v}_1 = \sigma_1 \boldsymbol{u}_1 = \begin{pmatrix} 0 \\ \sqrt{2} \end{pmatrix}, \quad \text{so} \quad \boldsymbol{u}_1 = \begin{pmatrix} 0 \\ 1 \end{pmatrix}; \\ A\boldsymbol{v}_2 &= \sigma_2 \boldsymbol{u}_2 \boldsymbol{v}_2^T \boldsymbol{v}_2 = \sigma_1 \boldsymbol{u}_2 = \begin{pmatrix} 2\sqrt{2} \\ 0 \end{pmatrix}, \quad \text{so} \quad \boldsymbol{u}_2 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}. \end{aligned}$$

The SVD of A is therefore

$$A = U\Sigma V^T = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} \sqrt{2} & 0 \\ 0 & 2\sqrt{2} \end{pmatrix} \begin{pmatrix} -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{pmatrix}.$$

Note that we can also compute the eigenvectors $\boldsymbol{u}_1$ and $\boldsymbol{u}_2$ directly from AA^T.

3 More Discussion

That $\text{rank}(A) = \text{rank}(\Sigma)$ tells us that we can determine the rank of A by counting the non-zero entries in Σ. In fact, we can do better. Recall that one of our complaints about Gaussian elimination was that it did not handle noise or nearly singular matrices well. SVD remedies this situation.

For example, suppose that an $n \times n$ matrix A is nearly singular. Indeed, perhaps A should be singular, but due to noisy data, it is not quite singular. This will show up in Σ, for instance, when all of the n diagonal entries in Σ are non-zero and some of the diagonal entries are almost zero.

More generally, an $m \times n$ matrix A may appear to have rank r, yet when we look at Σ we may find that some of the singular values are very close to zero. If there are l such values, then the "true" rank of A is probably $r - l$, and we would do well to modify Σ. Specifically, we should replace the l nearly zero singular values with zero.

Geometrically, the effect of this replacement is to reduce the column space of A and increase its null space. The point is that the column space is warped along directions for which $\sigma_i \approx 0$. In effect, solutions to $A\boldsymbol{x} = \boldsymbol{b}$ get pulled off to infinity (since $\frac{1}{\sigma_i} \approx \infty$) along vectors that are almost in the null space. So, it is better to ignore the ith coordinate by zeroing σ_i.

Example 2. The matrix

$$A = \begin{pmatrix} 1.01 & 1.00 & 1.00 \\ 1.00 & 1.01 & 1.00 \\ 1.00 & 1.00 & 1.00 \end{pmatrix}$$

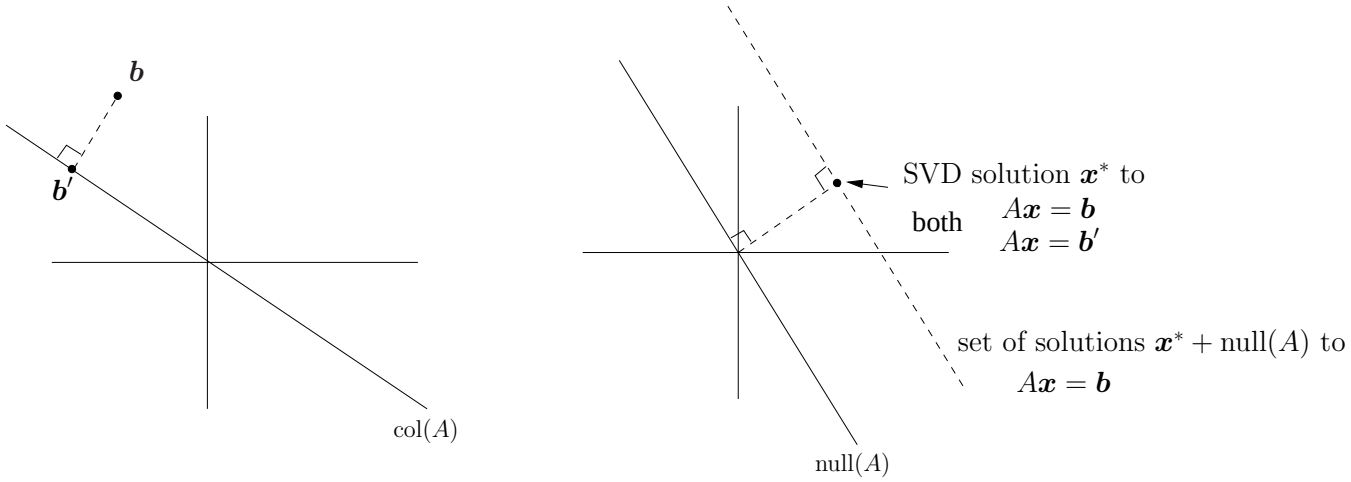
yields

$$\Sigma = \begin{pmatrix} 3.01 & 0 & 0 \\ 0 & 0.01 & 0 \\ 0 & 0 & 0.01 \end{pmatrix}.$$

Since 0.01 is significantly smaller than 3.01, we could treat it as zero and the rank A as one.

4 The SVD Solution

For the moment, let us suppose A is an $n \times n$ square matrix. The following picture sketches the way in which SVD solves the system $A\boldsymbol{x} = \boldsymbol{b}$.



The system $A\boldsymbol{x} = \boldsymbol{b}$ has an affine set of solutions, given by $\boldsymbol{x}_0 + \text{null}(A)$, where $\boldsymbol{x}_0$ is any solution. It is easy to describe the null space null(A) given the SVD decomposition, since it is just the span of the last $n - r$ columns of V. Also note that $A\boldsymbol{x} = \boldsymbol{b}$ has no solution since $\boldsymbol{b}$ is not in the column space of A.

SVD solves $A\boldsymbol{x} = \boldsymbol{b}$ by determining that $\boldsymbol{x}$ which is the closest to the origin, i.e., which has the minimum norm. It first projects $\boldsymbol{b}$ onto the column space of A, obtaining $\boldsymbol{b}'$, and then solves $A\boldsymbol{x} = \boldsymbol{b}'$. Essentially, SVD obtains the *least-squares* solution.

Given the diagonal matrix Σ in the SVD of A, denote by Σ^+ the diagonal matrix whose diagonal entries are of the form:

$$\left(\Sigma^+\right)_{ii} = \begin{cases} \frac{1}{\sigma_i} & \text{if } 1 \le i \le r; \\ 0 & \text{if } r+1 \le i \le n. \end{cases}$$

Thus the product matrix

$$\Sigma \cdot \Sigma^+ = \Sigma^+ \cdot \Sigma = \begin{pmatrix} 1 & & & & & \\ & \ddots & & & & \\ & & 1 & & & \\ & & & 0 & & \\ & & & & \ddots & \\ & & & & & 0 \end{pmatrix}$$

has r 1's on its diagonal. If Σ is invertible, then $\Sigma^+ = \Sigma^{-1}$.

Example 3. If

$$\Sigma = \begin{pmatrix} 2 & 0 & 0 \\ 0 & 5 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

then

$$\Sigma^+ = \begin{pmatrix} \frac{1}{2} & 0 & 0 \\ 0 & \frac{1}{5} & 0 \\ 0 & 0 & 0 \end{pmatrix}.$$

So how do we compute a solution to $A\boldsymbol{x} = \boldsymbol{b}$ for all the cases? We first compute the SVD decomposition $A = U\Sigma V^T$, then compute

$$\bar{\boldsymbol{x}} = V\Sigma^+ U^T \boldsymbol{b}.$$

(a) If A is invertible, then $\bar{\boldsymbol{x}}$ is the unique solution to $A\boldsymbol{x} = \boldsymbol{b}$.

(b) If A is singular and $\boldsymbol{b}$ is in the range of A, then $\bar{\boldsymbol{x}}$ is the solution closest to the origin. And $V\Sigma^+U^T$ is called a *pseudo-inverse* of A. The set of solutions is $\boldsymbol{x} + \text{null}(A)$, where null$(A)$ is spanned by the last $n - r$ columns of V.

(c) If A is singular and $\boldsymbol{b}$ is not in the range of A, then $\bar{\boldsymbol{x}}$ is the least-squares solution.

So far we have assumed that A is square. How do we solve $A\boldsymbol{x} = \boldsymbol{b}$ for general $m \times n$ matrix A? We still make use of the $n \times m$ pseudo-inverse matrix

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

The SVD solution is $\bar{\boldsymbol{x}} = A^+\boldsymbol{b}$.

The pseudoinverse A^+ has the following properties

$$\begin{aligned} A^+\boldsymbol{u}_i &= \begin{cases} \frac{1}{\sigma_i}\boldsymbol{v}_i & \text{if } i \le r, \\ 0 & \text{if } i > r; \end{cases} \\ \left(A^+\right)^T \boldsymbol{v}_i &= \begin{cases} \frac{1}{\sigma_i}\boldsymbol{u}_i & \text{if } i \le r, \\ 0 & \text{if } i > r. \end{cases} \end{aligned}$$

Namely, the vectors $\boldsymbol{u}_1, \ldots, \boldsymbol{u}_r$ in the column space of A go back to the row space. The other vectors $\boldsymbol{u}_{r+1}, \ldots, \boldsymbol{u}_m$ are in the null space of A^T, and A^+ sends them to zero. When we know what happens to each basis vector $\boldsymbol{u}_i$, we know A^+ because $\boldsymbol{v}_i$ and σ_i will be determined.

Example 4. The pseudoinverse of $A = \begin{pmatrix} 2 & 2 \\ -1 & 1 \end{pmatrix}$ from Example 1 is $A^+ = A^{-1}$, because A is invertible. And we have

$$A^+ = A^{-1} = V\Sigma^{-1}U^T = \begin{pmatrix} \frac{1}{4} & -\frac{1}{2} \\ \frac{1}{4} & \frac{1}{2} \end{pmatrix}.$$

In general there will not be any zero singular values. However, if A has column degeneracies there may be near-zero singular values. It may be useful to zero these out, to remove noise. (The implication is that the overdetermined set is not quite as overdetermined as it seemed, that is, the null space is not trivial.)

5 SVD Algorithm

Here we only take a brief look at how the SVD algorithm actually works. For more details we refer to [2]. It is useful to establish a contrast with Gaussian elimination, which reduces a matrix A by a series of row operations that zero out portions of columns of A. Row operations imply pre-multiplying the matrix A. They are all collected together in the matrix L^{-1} where $A = LDU$.

In contrast, SVD zeros out portions of both rows and columns. Thus, whereas Gaussian elimination only reduces A using pre-multiplication, SVD uses both pre- and post-multiplication. As a result, SVD can at each stage rely on orthogonal matrices to perform its reductions on A. By using orthogonal matrices, SVD reduces the risk of magnifying noise and errors. The pre-multiplication matrices are gathered together in the matrix U^T, while the post-multiplication matrices are gathered together in the matrix V.

There are two phases to the SVD decomposition algorithm:

(i) SVD reduces A to bidiagonal form using a series of orthogonal transformations. This phase is deterministic and has a running time that depends only on the size of the matrix A.

(ii) SVD removes the superdiagonal elements from the bidiagonal matrix using orthogonal transformation. This phase is iterative but converges quickly.

Let us take a slightly closer look at the first phase. Step 1 in this phase creates two orthogonal matrices U_1 and V_1 such that

$$U_1 A = \begin{pmatrix} a'_{11} & a'_{12} & \cdots & a'_{1n} \\ 0 & & & \\ \vdots & & B' & \\ 0 & & & \end{pmatrix},$$

$$U_1 A V_1 = \begin{pmatrix} a''_{11} & a''_{12} & 0 & \cdots & 0 \\ 0 & & & & \\ \vdots & & B'' & & \\ 0 & & & & \end{pmatrix}.$$

If A is $m \times n$ then B'' is $(m-1) \times (n-1)$. The next step of this phase recursively works on B'', and so forth, until orthogonal matrices $U_1, \ldots, U_{n-1}, V_1, \ldots, V_{n-2}$ are produced such that $U_{n-1} \cdots U_1 A V_1 \cdots V_{n-2}$ is bidiagonal (assume $m \geq n$).

In both phases, the orthogonal transformation are constructed from Householder matrices. For practitioners, free online libraries such as [5, 6] offer SVD source code in C++.

References

[1] M. Erdmann. Lecture notes for *16-811 Mathematical Fundamentals for Robotics*. The Robotics Institute, Carnegie Mellon University, 1998.

[2] G. E. Forsythe, M. A. Malcolm, and C. B. Moler. *Computer Methods for Mathematical Computations*. Prentice-Hall, 1977.

[3] W. H. Press, *et al. Numerical Recipes in C++: The Art of Scientific Computing.* Cambridge University Press, 2nd edition, 2002.

[4] G. Strang. *Introduction to Linear Algebra.* Wellesley-Cambridge Press, 1993.

[5] Boost C++ Libraries. `http://www.boost.org/`.

[6] LIT-lib. `http://ltilib.sourceforge.net/doc/homepage/index.shtml`.