7 Matrix Decompositions

Topics:

1. Introduction
2. Vector Space Basics
   - Definition of a Vector Space
   - Norms
   - Linear Combinations
   - Inner products
   - Change of Bases
3. Linear Mappings
4. The SVD decomposition
5. Applications of the SVD

7.1 Introduction

The primary purpose of this section is to present the singular value decomposition (SVD), a spectacularly useful way to write any matrix and one that has manifold applications in science, technology, and computation. We begin the section by reviewing (and perhaps extending) some things you learned in linear algebra, including norms, vectors, and matrices, and lay the groundwork for viewing matrices as mappings, i.e. as entities that “do things”, rather than simply passively exist.

7.2 Vector Space Basics

One of the driving ideas in this course has been the idea of bounding or at least quantifying errors. In order to talk sensibly about errors, however, we need to have a notion of size, i.e. to be able to say which quantities are big and which ones are small. This task is not as simple as it might appear to be, however, for the simple reason that the set of “things” that might be computed is vast and heterogenous: it includes numbers, vectors, matrices, and functions, among many other things. While absolute error and relative error might be natural concepts for numbers, how do they extend, e.g., to functions?

As luck would have it, most of the objects we are concerned about in this course can be viewed as living in vector spaces, a setting in which the concept of a norm provides a clean way to think about size.

7.2.1 Definition of a vector space

Recall that a vector space is a set whose elements can be added together and multiplied by scalars (to yield new elements that are also in the set.) More formally, a vector space over the real numbers is a set $X$ together with two mappings, one called addition (mapping $X \times X \rightarrow X$) and the other called scalar multiplication (mapping $\mathbb{R} \times X \rightarrow X$.) These mappings must satisfy the following familiar properties for all $x, y, z$ in $X$ and $a,b$ in $\mathbb{R}$:

1. $x + (y + z) = (x + y) + z$
2. $x + y = y + x$
3. $\exists$ an element $0$ such that $0 + x = x$
4. $\exists$ an element $-x$ such that $-x + x = 0$
5. $a(x + y) = ax + ay$.
6. $(a + b)x = ax + bx$.
7.2 Vector Space Basics

7. \( a(bx) = (ab)x \)

8. \( 1x = x \)

Below are a few important examples of vector spaces.

**Example 7.1. (Vectors)** The set \( \mathbb{R}^n \), i.e. the set of objects of the form \((x_1, \cdots, x_n)\), where addition is performed element-wise, i.e.

\[
(x_1, \cdots, x_n) + (y_1, \cdots, y_n) = (x_1 + y_1, \cdots, x_n + y_n)
\]

and scalar multiplication satisfies

\[
c(x_1, \cdots, x_n) = (cx_1, \cdots, cx_n).
\]

**Example 7.2. (Matrices)** The set \( \mathbb{R}^{n \times m} \), i.e. the set of \( n \times m \) matrices with real elements. As with the previous example, addition is performed element-wise and scalar multiplication distributes the scalar to every element.

**Example 7.3. (Functions)** The set of real-valued continuous functions on the interval \([0, 1]\), where addition satisfies

\[
(f + g)(x) = f(x) + g(x)
\]

and multiplication satisfies

\[
(cf)(x) = c(f(x)).
\]

### 7.2.2 Norms

Given a vector space \( V \), a norm on \( V \) is a mapping \( \rho : V \to [0, \infty) \) satisfying the following properties:

1. \( \rho(ax) = |a|\rho(x) \)
2. \( \rho(x + y) \leq \rho(x) + \rho(y) \)
3. If \( \rho(x) = 0 \), then \( x = 0 \).

Norms provide a notion of size: one we find a norm \( \rho \), we can say that element \( x \) is bigger than element \( y \) if \( \rho(x) > \rho(y) \). Similarly, we can talk about how far apart two elements are by considering the norm of their difference: \( x \) and \( y \) are close if \( \rho(x - y) \) is small, far if it is large.

Here are some examples of norms:

**Example 7.4. (Scalars)** For \( x \in \mathbb{R} \), the most familiar norm is that of the absolute value, defined formally as

\[
|x| = \begin{cases} 
  x & x \geq 0 \\
  -x & x < 0.
\end{cases}
\]

**Example 7.5. (Vectors)** Let \( x \) be an \( n \)-dimensional vector, i.e. \( x = (x_1, x_2, \cdots, x_n) \in \mathbb{R}^n \). The \( p \)-norm of \( x \) is defined as

\[
\|x\|_p = \left( \sum_{i=1}^{n} x_i^p \right)^{1/p}, \quad 1 \leq p < \infty.
\]

There is a variant for \( p = \infty \) as well, called the sup-norm:

\[
\|x\|_\infty = \max_{i=1,\cdots,n} |x_i|.
\]

Note that in the special case that \( n = 1 \), i.e. in the case where \( x \) is a scalar, both the \( p \)-norms and sup-norm reduce to just the absolute value of \( x \).
Example 7.6. (Matrices:) Let $A$ be a matrix with $n$ rows and $m$ columns, i.e.

$$
A = \begin{pmatrix}
    a_{11} & \cdots & a_{1m} \\
    \vdots & \ddots & \vdots \\
    a_{n1} & \cdots & a_{nm}
\end{pmatrix} \in \mathbb{R}^{n \times m}.
$$

Recall from linear algebra that the matrix-vector product $y = Ax$ is defined for any $x \in \mathbb{R}^m$, and that in this case $y \in \mathbb{R}^n$. In other words, $A$ is a mapping which takes elements $x \in \mathbb{R}^m$ to elements $y \in \mathbb{R}^n$. The size of $A$ is generally defined as the maximum of the ratios of the size of $y$ to the size of $x$, i.e.

$$
\|A\| = \sup_{x \in \mathbb{R}^m} \frac{\|Ax\|}{\|x\|}.
$$

Note that the numerator of this expression requires a notion of size in the vector space $\mathbb{R}^n$, while the denominator uses size in $\mathbb{R}^m$. As noted above, there are multiple options for how to define size in these vector spaces. If the sizes of $x$ and $y$ are measured in the same $p$-norm, then the induced matrix norm is generally notated with a subscript $p$:

$$
\|A\|_p = \sup_{x \in \mathbb{R}^m} \frac{\|Ax\|_p}{\|x\|_p}.
$$

You can conceive of mixed matrix norms, too, where the size of the input is measured in a $p$-norm and the size of the output is measured in some other norm, e.g. a $q$-norm, $q \neq p$. In this case, we have

$$
\|A\|_{p,q} = \sup_{x \in \mathbb{R}^m} \frac{\|Ax\|_q}{\|x\|_p}.
$$

These matrix norms are rather exotic and not commonly used in applications.

Example 7.7. (Functions:) Let $f(x)$ be a function with domain $\mathbb{R}$. Just as with vectors, there is a family of $p$-norms that can be used to measure the size of $f$:

$$
\|f\|_p = \left( \int_{-\infty}^{\infty} |f(x)|^p dx \right)^{1/p}, \quad 1 \leq p < \infty.
$$

Note that not every function $f(x)$ defined on $\mathbb{R}$ will have a finite $p$ norm, and that there are functions that have a finite $p$-norm for certain values of $p$ and not for others. Also note that if both $f$ and $g$ have finite $p$ norm for some particular choice of $p$, then so does the function $h(x) = \alpha f(x) + \beta g(x)$ (exercise for the reader.)

Just as with vectors, the sup-norm for functions is defined slightly differently:

$$
\|f\|_{\infty} = \sup_{x \in \mathbb{R}} |f(x)|,
$$

where sup represents the supremum.

7.2.3 Bases

Suppose $v_1, \cdots, v_m$ are vectors in a vector space $V$. A linear combination of the $v_i$ is a sum of the form

$$
\alpha_1 v_1 + \cdots + \alpha_m v_m,
$$

where the $\alpha_i$ are all scalars. The span of the $v_i$ is the set of all linear combinations of those vectors. It is easy to see that the span of any set of vectors forms a subspace $S$, i.e. a collection of vectors that is closed under addition and scalar multiplication. We refer to such a subspace as the subspace generated by the $v_i$.

If the subspace generated by a set of vectors $v_i$ is all of $V$, the $v_i$ are called a spanning set of $V$. (Similarly, a spanning set for a subspace $S$ is any set whose span is precisely $S$. ) A spanning set is called
minimal if there is no smaller collection of vectors with the same span. A basis for \( V \) or \( S \) is any minimal spanning set for these spaces, and the dimension of \( V \) or \( S \) is the number of elements in their basis.

A set of vectors \( v_1, \cdots, v_n \) is called linearly independent if no element of the set is a linear combination of the others, or, equivalently, if the only way for the equation

\[
\alpha_1 v_1 + \cdots + \alpha_n v_n = 0
\]

to be true is for the coefficients \( \alpha_i \) all to equal 0. It is easy to see that a minimal spanning set must be linearly independent, so an equivalent definition for a basis is any linearly independent spanning set.

Example 7.8. Consider vectors \( v_1, v_2, v_3, v_4 \in \mathbb{R}^2 \), with \( v_1 = (1, 0) \), \( v_2 = (1, 1) \), \( v_3 = (0, 1) \), and \( v_4 = (2, 2) \). Check that the span of the \( v_i \) is all of \( \mathbb{R}^2 \). The \( v_i \) do not form a basis, however, as \( v_4 = 2v_2 \), \( v_2 = v_1 + v_3 \), i.e. the vectors are not linearly independent. Subsets of the \( v_i \) that comprise bases include the sets \( \{v_1, v_3\} \), \( \{v_1, v_2\} \), and \( \{v_2, v_3\} \). The set \( \{v_2, v_4\} \) does not comprise a basis, however: the span of this collection is a one dimensional subspace of \( \mathbb{R}^2 \).

7.2.4 Inner Products

Given a vector space \( V \), an inner product on \( V \) is a bilinear mapping \( < \cdot, \cdot > : V \times V \to \mathbb{R} \), i.e. a mapping that satisfies

\[
< \alpha v + \beta w, z > = \alpha < v, z > + \beta < w, z >, \quad < v, \alpha w + \beta z > = \alpha < v, w > + \beta < v, z >
\]

for any vectors \( v, w, \) and \( z \) in \( V \) and any scalars \( \alpha \) and \( \beta \) in \( \mathbb{R} \). In general, there are many ways to define an inner product on any given vector space.

Example 7.9. If \( v = (v_1, \cdots, v_n) \) and \( w = (w_1, \cdots, w_n) \) are vectors in \( \mathbb{R}^n \), then the canonical inner product between \( v \) and \( w \) is defined as

\[
<v, w> = \sum_{i=1}^{n} v_i w_i.
\]

You could define an alternative inner product as

\[
<v, w>_a = \sum_{i=1}^{n} \alpha_i v_i w_i,
\]

where the \( \alpha_i \) are small positive numbers. (Exercise for the reader: check that \( < \cdot, \cdot >_a \) is actually an inner product.)

Example 7.10. Let \( C[0,1] \) denote the set of continuous functions on the closed interval \([0,1]\). Then we can define an inner product as

\[
<f, g> = \sup_x |f(x)g(x)|.
\]

Note that the usual norm of a vector can be given in terms of its canonical inner product as

\[
\|v\|^2 = <v, v>.
\]

A vector whose norm is 1 is called normal.

Two vectors \( v \) and \( w \) are called orthogonal if their inner product satisfies \( <v, w> = 0 \). A basis is called an orthogonal basis if its elements are all mutually orthogonal, and it is called an orthonormal basis if its elements are both normal and orthogonal to one another.
7.2.5 Vector representations

Given any basis \(v_1, \ldots, v_n\) for \(\mathbb{R}^n\), any vector \(x \in \mathbb{R}^n\) can be represented in a unique way as a linear combination of the \(v_i\), i.e.
\[
x = \alpha_1 v_1 + \cdots + \alpha_n v_n.
\]
(42)

If the \(v_i\) happen to be orthogonal, the coefficients of this representation are given by the formula
\[
\alpha_i = \frac{\langle x, v_i \rangle}{\|v_i\|}.
\]
(43)

If the \(v_i\) are not orthogonal, this simple formula needs to be replaced by the somewhat more complicated approach. In \(\mathbb{R}^n\), this approach boils down to matrix inversion, as the following example shows.

Example 7.11 (Change from standard basis to some other basis). Suppose \(x = (x_1, \ldots, x_n)\) is a vector in \(\mathbb{R}^n\), and we want to calculate the coefficients of \(x\) with respect to a basis \(v_1, \ldots, v_n\), where \(v_i = (v_{i1}, \ldots, v_{in})\). Then equation (42) is a matrix equation of the form
\[
M \alpha = x,
\]
where \(M\) is the matrix whose columns are just the vectors \(v_i\), i.e.
\[
M = [v_1| \cdots |v_n] = \begin{pmatrix} v_{11} & \cdots & v_{1n} \\ \vdots & \vdots & \vdots \\ v_{n1} & \cdots & v_{nn} \end{pmatrix}
\]
and \(\alpha\) is the vector of coefficients, \(\alpha = (\alpha_1, \cdots, \alpha_n)^T\). The coefficients are then given via the matrix inversion
\[
\alpha = M^{-1} x.
\]
(44)

Note that if the \(v_i\) happen to be orthogonal, then \(M^{-1}\) is given by
\[
M^{-1} = \begin{pmatrix} v_1^T / \|v_1\| \\ \vdots \\ v_n^T / \|v_n\| \end{pmatrix},
\]
and the \(i\)th component of \(M^{-1} x\) is exactly the expression given in (43).

Example 7.12 (Change of arbitrary basis to some other basis). Suppose you already know the coefficients of \(x\) with respect to basis \(v_1, \ldots, v_n\), and you wish to know them in terms of a different basis \(u_1, \ldots, u_n\). Then we define two matrices,
\[
M = [v_1| \cdots |v_n], \quad N = [u_1| \cdots |u_n].
\]
If \(x = \alpha_1 v_1 + \cdots + \alpha_n v_n\), then \(M \alpha\) simply expresses \(x\) in terms of the standard basis, whence, as in the last example, \(N^{-1}(M \alpha)\) represents \(x\) in terms of \(u\).

7.3 Linear mappings

A linear mapping \(A : \mathbb{R}^n \to \mathbb{R}^m\) is any function satisfying
\[
A(\alpha_1 v_1 + \alpha_2 v_2) = \alpha_1 Av_1 + \alpha_2 Av_2
\]
(45)
for \(v_1\) and \(v_2\) in \(\mathbb{R}^n\) and \(\alpha_1\) and \(\alpha_2\) in \(\mathbb{R}\). Suppose \(v_1, \ldots, v_n\) is a basis for \(\mathbb{R}^n\), and we happen to know that linear mapping \(A\) satisfies \(Av_i = u_i\) for some vectors \(u_i\). If we know the \(u_i\), then we can figure out where \(A\)
takes any vector \( x \in \mathbb{R}^n \), since by (42) we can write \( x = \alpha_1 v_1 + \cdots + \alpha_n v_n \), whereupon by (45) we have

\[
Ax = A \sum_{i=1}^{n} \alpha_i v_i \\
= \sum_{i=1}^{n} \alpha_i A v_i \\
= \sum_{i=1}^{n} \alpha_i u_i.
\]

In other words, the action of \( A \) is completely specified by its action on a basis.

### 7.3.1 Matrix representations of linear mappings

We often see linear mappings \( A : \mathbb{R}^n \to \mathbb{R}^m \) represented as matrices. How we choose to represent the mapping depends on our choice of basis. The relation between the action of the map \( A \) and the matrix that is used to represent it is as follows: first, choose two bases, one (denoted \( v_1, \cdots, v_n \)) for the domain space \( \mathbb{R}^n \) and one (denoted \( u_1, \cdots, u_m \)) for the range space \( \mathbb{R}^m \). Using (43) or (44), calculate numbers \( a_{ij} \) such that

\[
Av_j = \sum_{i=1}^{m} a_{ij} u_i,
\]

i.e. calculate the coefficients of the vector \( Av_j \) with respect to the basis \( u_i \). Then the matrix representation of \( A \) is given by

\[
A = \begin{pmatrix}
  a_{11} & \cdots & a_{1n} \\
  \vdots & \ddots & \vdots \\
  a_{m1} & \cdots & a_{mn}
\end{pmatrix}
\]

Note that a single linear mapping \( A \) will have infinitely many matrix representations (one for each choice of basis.)

**Example 7.13 (Rotation matrices).** A rotation matrix rotates vectors by some fixed angle around the origin, leaving the size of the vector totally unchanged. In \( \mathbb{R}^2 \), a matrix representation with respect to the standard basis of “rotation by \( \theta \) degrees” is

\[
R_{\theta} = \begin{pmatrix}
  \cos \theta & -\sin \theta \\
  \sin \theta & \cos \theta
\end{pmatrix}
\]

Check that \( R_{\theta} \) does in fact rotate vectors by angle \( \theta \).

**Example 7.14 (Projection Matrices).** To project a vector onto a subspace means to find the element in the subspace that is “closest” to the input vector. For example, if \( S \) is the subspace of elements in \( \mathbb{R}^2 \) of the form \((x, 0)\), then projection of a vector \((x, y)\) onto \( S \) amounts to replacing \( y \) with 0. A matrix representation of projection onto this subspace is given by

\[
P = \begin{pmatrix}
  1 & 0 \\
  0 & 0
\end{pmatrix}
\]

Note again that this representation is with respect to the standard basis.

### 7.3.2 Diagonalization of matrices

If with respect to bases \( v_i \) and \( u_j \) a matrix \( A \) is diagonal, i.e.

\[
A = \begin{pmatrix}
  \sigma_1 & 0 & \cdots & 0 \\
  0 & \sigma_2 & \ddots & \vdots \\
  \vdots & \ddots & \ddots & 0 \\
  0 & 0 & \cdots & \sigma_n
\end{pmatrix},
\] (46)
then this means that $A$ takes $v_i$ to $\sigma_i u_i$ for all indices $i$. Note that if you have diagonal matrix $A$ of the form (46), it’s easy to calculate the action of both $A$ and $A^{-1}$ on any vector $x$. If $x = \alpha_1 v_1 + \cdots + \alpha_n v_n$, then $Ax$ is just

$$Ax = \sum_{i=1}^{n} \alpha_i \sigma_i u_i$$

and, similarly, if $y = \beta_1 u_1 + \cdots + \beta_n u_n$, then $A^{-1}y$ is

$$A^{-1}y = \sum_{i=1}^{n} \frac{\beta_i}{\sigma_i} v_i.$$

**Example 7.15.** Consider a $2 \times 2$ matrix $A$ with eigenvalues $\lambda_1 = 2$ and $\lambda_2 = 1$ and eigenvectors $x_1 = (1, 0)$ and $x_2 = (1/2, \sqrt{3}/2)$. Note that since $x_2$ is not a multiple of $x_1$, the set $\{x_1, x_2\}$ is a basis for $\mathbb{R}^2$. If we choose this basis for both the domain and range of $A$, then $A$ has the representation

$$A = \begin{pmatrix} 2 & 0 \\ 0 & 1 \end{pmatrix},$$

i.e. $A$ is diagonal. *(This is the case for any matrix $A$, expressed relative to an eigenbasis.)*

Suppose we wanted to represent $A$ with respect to the standard basis $y_1 = (1, 0)$, $y_2 = (0, 1)$. Define

$$M = [x_1|x_2] = \begin{pmatrix} 1 & 1/2 \\ 0 & \sqrt{3}/2 \end{pmatrix}$$

The the matrix

$$MAM^{-1} = \begin{pmatrix} 2 & -0.57735 \\ 0 & 1 \end{pmatrix}$$

acts by first changing from the standard basis to the basis $x_1, x_2$, then applying $A$ (in the form developed above), then changing back to the standard basis.

### 7.4 The Singular Value Decomposition

Having seen the utility of diagonalization, and also of using orthogonal bases, it is natural to ask a few questions:

- Which linear mappings can be diagonalized relative to a single orthogonal basis? (i.e. using the same orthogonal basis for both the domain and range?)
- Which linear mappings can be diagonalized relative to *some* single basis? (i.e. using the same basis for both the domain and range?)
- Which linear mappings can be diagonalized relative to two different orthogonal bases? (i.e. using one orthogonal basis for the domain and a different one for the range?)

These questions have nice answers. The set of linear mappings that can be diagonalized relative to a single orthogonal basis is just the set of *symmetric* matrices, i.e. matrices $A$ for which $A^T = A$. This fact is not totally transparent, but you probably saw it in your first course in linear algebra. On the other hand, if you believe that every square matrix of size $n \times n$ has $n$ eigenvalues (counting multiplicity) and $n$ eigenvectors, then it is obvious that every matrix can be diagonalized relative to *some* basis (in particular, relative to its eigenvectors.) The problem with this characterization is that it is generally much easier to work with orthogonal bases, not with arbitrary basis. It is an amazing fact that it is also the case that *any* matrix (not just square ones) can be diagonalized relative to two (possibly different) orthogonal bases. This is the content of the Singular Value Decomposition:

**Theorem 7.1 (Singular Value Decomposition).** Suppose $m \geq n$, and $A : \mathbb{R}^n \to \mathbb{R}^m$ is a linear mapping. Then there exist positive numbers $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_n \geq 0$ and unitary matrices $U \in \mathbb{R}^{m \times n}$ and $V \in \mathbb{R}^{n \times n}$ such that

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

(47)
7.5 Applications of the SVD

where $\Sigma \in \mathbb{R}^{m \times n}$ is a “diagonal” matrix of the form

$$
\Sigma = \begin{pmatrix}
\sigma_1 & \cdots & 0 \\
\vdots & \ddots & \vdots \\
0 & \cdots & \sigma_n \\
\vdots & \ddots & \vdots \\
0 & \cdots & 0
\end{pmatrix}
$$

(Comment: the condition $m \geq n$ is not essential; we include it here just because it streamlines notation. If $m \leq n$, then there will be $m$ singular values, and the excess zeros in $\Sigma$ will extend off to the right rather than towards the bottom.)

Some things to note:

- The columns of $V$ form an orthogonal basis for the domain of $A$.
- The columns of $U$ form an orthogonal basis for the range of $A$.
- Saying that $AV = \Sigma U$ that $A v_i = \sigma_i u_i$, $1 \leq i \leq n$,
  i.e. that $A$ is diagonal with respect to domain basis $\{u_i\}$ and range basis $\{v_i\}$.
- If $m > n$, then the range of $A$, denoted $\text{Ran}(A)$, is a strict subset of $\mathbb{R}^m$. (This follows from the fact that the dimension of the kernel plus the dimension of the range is equal to the dimension of the domain of $A$.) The subspace $\mathbb{R}^m / \text{Ran}(A)$ (i.e. the complement of the subspace $\text{Ran}(A)$) is spanned by the $n + 1$st through the $m$th column of $U$. In fact, these columns could be swapped out for any orthonormal basis of $\mathbb{R}^m / \text{Ran}(A)$, and the resulting $U$ would still be unitary and still satisfy (47). In other words, $U$ is not unique!
- If some $\sigma_i = 0$, then the subspace of $\mathbb{R}^n$ spanned by $v_i$ lies in the kernel of $A$.

7.5 Applications of the SVD

As we’ve already pointed out, one of the chief advantages of diagonalization is that it makes it very easy to compute the action of both $A$ and $A^{-1}$. For our immediate purposes, the value of these computations is that it will make it easy for us to solve the problem

$$
\min_{x \in \mathbb{R}^n} \|Ax - y\|. \quad (48)
$$

Our our interest in this problem is directly related to our interest in deblurring images, as will become clear below. The remainder of this section is devoted to explaining the relation between the SVD formula (47), the problem (48), and the image deblurring problem.

7.5.1 The Pseudo-Inverse

Suppose $m > n$ and $A : \mathbb{R}^n \to \mathbb{R}^n$ is a linear mapping. Then some vectors $y \in \mathbb{R}^m$ will lie in the range of $A$, but most will not. Even if $y \in \text{Ran}(A)$, however, it is possible that $A$ is not injective, i.e. that $y$ is the image of multiple vectors in the domain space $\mathbb{R}^n$. In spite of these complications, we would like to be able to talk intelligibly the “inverse” of any $y$ under $A$. The pseudo-inverse allows us to do this: it defines, for each $y \in \mathbb{R}^m$, a single $x \in \mathbb{R}^n$ with two important properties:

- $Ax$ is “as close to $y$ as possible”.
- $x$ itself is the smallest vector that satisfies this property.

These concepts will be made precise below.
Definition 7.1. [The Pseudoinverse] Let \( m \geq n \) and \( A : \mathbb{R}^n \to \mathbb{R}^m \) be a linear mapping with SVD decomposition \( U \Sigma V^T \). For any \( y \in \mathbb{R}^m \), the pseudo-inverse of \( y \) is defined as

\[
x^\dagger = V \Sigma^\dagger U^T y
\]

where \( \Sigma^\dagger \) is defined as the matrix

\[
\Sigma^\dagger = \begin{pmatrix}
\sigma_1^\dagger & \cdots & 0 \\
\vdots & \ddots & \vdots \\
0 & \cdots & \sigma_n^\dagger \\
\vdots & \ddots & \vdots \\
0 & \cdots & 0
\end{pmatrix}
\]

whose diagonal entries are given by

\[
\sigma_i^\dagger = \begin{cases}
\frac{1}{\sigma_i} & \sigma_i \neq 0 \\
0 & \sigma_i = 0.
\end{cases}
\]

Some things to note about the pseudo-inverse:

- If \( m = n \) and none of the \( \sigma_i = 0 \), then the pseudo-inverse and the inverse coincide.

- Recall that if \( \sigma_i = 0 \) for some \( i \), then the subspace spanned by \( u_i \) belongs to the kernel of \( A \), whence the subspace spanned by \( v_i \) is orthogonal to the range of \( A \). If some \( y \) has a component along \( v_i \), the pseudo-inverse simply zeros this component out, and takes the (pseudo-)inverse of what’s left over. In other words, in projects the vector \( y \) onto the range of \( A \), and then takes the inverse.

7.5.2 The relation between the pseudo-inverse and the least squares problem

Theorem 7.2. The quantity \( x^\dagger \) defined in (6.1) solves the minimization problem given in (48).

Proof. This proof is an exercise that will show up on your first take home exam. As a way to get started, however, start with an arbitrary \( y \in \mathbb{R}^m \), and write it as

\[
y = \sum_{i=1}^{m} b_i u_i,
\]

where \( u_i \) is the \( i \)th column of \( U \). Then consider an arbitrary \( x \in \mathbb{R}^n \), and write this \( x \) as

\[
x = \sum_{i=1}^{m} a_i v_i.
\]

Now calculate the norm of \( \|Ax - y\| \), remembering the following:

- \( Av_i = \sigma_i u_i \)
- The \( u_i \) are mutually orthogonal
- That \( \|w + z\|^2 = \|w\|^2 + \|z\|^2 \) for any orthogonal vectors \( w \) and \( z \).

Which choice of \( a_i \) yields the smallest norm?

7.5.3 Truncated SVD Algorithm

It is illuminating to note that an alternative way of writing \( x^\dagger \) is as

\[
x^\dagger = \sum_{i=1}^{n} \frac{u_i^T y}{\sigma_i} v_i.
\]
In other words, we can think of \( x^\dagger \) as the sum of \( n \) terms, each a scalar multiple of one of the columns of \( V \). For small but non-zero values of \( \sigma_i \), these scalar multiples become intractable. The idea behind the **Truncated SVD Algorithm** is to approximately solve (48) by taking only the terms of this sum that are well behaved.

**Truncated SVD algorithm:**

Suppose \( A \) has singular values \( \sigma_1 \geq \cdots \geq \sigma_n \). To approximate the solution to (48), proceed as follows:

1. Fix a threshold \( \epsilon > 0 \).
2. Set \( p = \max_i \{ \sigma_i > \epsilon \} \).
3. Define the **truncated SVD solution** as
   \[
   x^\dagger_{svd} = \sum_{i=1}^{p} \frac{u_i^T y}{\sigma_i} v_i. \tag{50}
   \]
4. If you don’t like this solution, change \( \epsilon \) and try again.

### 7.5.4 Tikhonov Regularization

The idea behind the SVD truncation algorithm is that you solve an “ill-posed” problem approximately. The idea behind Tikhonov regularization is actually to change the problem, so that you solve exactly a problem that is a “pretty good” approximation of the original. Formally, the Tikhonov procedure is to solve the problem

\[
\min_x \| Ax - y \|_2^2 + \alpha \| x \|_2^2, \tag{51}
\]

where \( \alpha \) is some positive parameter. In general, \( \alpha \) is chosen, i.e. if you wish to use the Tikhonov procedure, you must specify some particular value of \( \alpha \). The hope is that the \( \alpha \) you choose will yield a “good” solution. There are principled techniques for choosing good \( \alpha \)’s, but in many practical cases choosing \( \alpha \) is a matter of trial and error.

It turns out that you can use the SVD to solve (51), just as you could use it to solve (47). To see this, note that (51) is equivalent to

\[
\min_x \left\| \left( \begin{array}{cc} A & \alpha I \end{array} \right) x - \left( \begin{array}{cc} y \\ 0 \end{array} \right) \right\|_2^2, \tag{52}
\]

and that (52) is simply a least squares problem involving the augmented matrices \([A \quad \alpha I]^T\) and \([y \quad 0]^T\).

It is a cute exercise (and one you should do) to show that the singular values of the augmented matrix \([A \quad \alpha I]^T\) are simply

\[
\hat{\sigma}_i = \sigma_i + \alpha, \quad i = 1 \cdots n,
\]

where \( \sigma_i \) are the singular values of \( A \). Thus the augmented system is well-conditioned, and the SVD algorithm can be used to solve (52), where in effect \( \alpha \) replaces the parameter \( \epsilon \).

**Tikhonov algorithm:**

1. Choose an \( \alpha \)
2. Solve (51) exactly (one way to do this is to use the SVD algorithm on (52).)
3. If you don’t like your solution, change \( \alpha \).
7.5 Applications of the SVD

7.5.5 Application to Deblurring

For our purposes, a digital image is an $m \times n$ matrix of real numbers. Each number represents an intensity value for light in an image plane. Each coordinate pair $(i,j)$, $1 \leq i \leq m$, $1 \leq j \leq n$, is called a pixel. Sometimes we will also think of images as vectors. To pass from the matrix representation of an image to the vector representation of an image, simply rearrange the pixels into a single column by stacking matrix columns one on top of the other.

A blurred image is an image whose pixels have been replaced by a (weighted) average of nearby pixels. Blurring is a linear operation. If we view an image as a vector, we can view a blurring kernel as a matrix.

**Example 7.16.** Consider a very simple $3 \times 3$ image, i.e. a matrix of the form

$$I = \begin{pmatrix} a & d & g \\ b & e & h \\ c & f & i \end{pmatrix},$$

where $a, \ldots, i$ are real numbers. Suppose $I$ is blurred by replacing each pixel by the average of its immediate neighbors (excluding diagonals.) Then the blurring kernel and the vectorized image are given by

$$K = \begin{pmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{pmatrix}, \quad \text{and} \quad \hat{I} = \begin{pmatrix} a \\ b \\ c \\ d \\ e \\ f \\ g \\ h \\ i \end{pmatrix},$$

respectively. Note that the blurring kernel is very large and also very structured.

Since the blurred image is given by

$$\hat{I}_{\text{blur}} = K\hat{I},$$

the unblurred image would be given by

$$\hat{I} = K^{-1}\hat{I}_{\text{blur}}$$

if $K$ were invertible. In general, $K$ is not invertible, so we seek a solution to the problem

$$\min_{\hat{I}} \|K\hat{I} - \hat{I}_{\text{blur}}\|.$$  \hspace{1cm} (53)

This is exactly the kind of problem we can use the SVD truncation and Tikhonov algorithms for.