

Linear algebra for computer vision

Bharath Hariharan

January 15, 2020

1 Vector spaces

Definition 1 A *vector space* V is a nonempty set of objects \mathbf{v} , with two operations defined on them: multiplication by a scalar c (belonging to a field; here let's assume c is a real number), denoted as $c\mathbf{v}$, and addition of two vectors, denoted as $\mathbf{u} + \mathbf{v}$ that satisfy the following properties:

1. The vector space is closed under both addition and scalar multiplication. That is, if $\mathbf{u}, \mathbf{v} \in V$ and c is a real number, then $c\mathbf{u} \in V$ and $\mathbf{u} + \mathbf{v} \in V$.
2. Vector addition is commutative and associative. That is, $\mathbf{u} + \mathbf{v} = \mathbf{v} + \mathbf{u}$, and $\mathbf{u} + (\mathbf{v} + \mathbf{w}) = (\mathbf{u} + \mathbf{v}) + \mathbf{w}$ for all $\mathbf{u}, \mathbf{v} \in V$.
3. There is a zero vector $\mathbf{0} \in V$ s.t $\mathbf{u} + \mathbf{0} = \mathbf{u}$ for all $\mathbf{u} \in V$.
4. For every vector $\mathbf{u} \in V$, there exists $-\mathbf{u} \in V$ s.t $\mathbf{u} + (-\mathbf{u}) = \mathbf{0}$.
5. Scalar multiplication distributes over addition: $c(\mathbf{u} + \mathbf{v}) = c\mathbf{u} + c\mathbf{v}$, and $(c + d)\mathbf{u} = c\mathbf{u} + d\mathbf{u}$.
6. $c(d\mathbf{u}) = (cd)\mathbf{u}$.
7. $1\mathbf{u} = \mathbf{u}$.

A vector space is best thought of as a generalization of the cartesian plane. Consider the cartesian plane, which is the set of all points (x, y) , where x and y are real numbers. Define addition to be element-wise addition: $(x, y) + (x', y') = (x + x', y + y')$. Similarly, define scalar multiplication to be element-wise: $c(x, y) = (cx, cy)$. Define the zero vector to be $(0, 0)$. For $\mathbf{u} = (x, y)$, define $-\mathbf{u} = (-1)\mathbf{u} = (-x, -y)$. Test each of the properties described above and make sure that they are indeed true.

Points (x, y) in the cartesian plane can be thought of in computer science parlance as numeric arrays of size 2. We can in fact produce a more general example by considering the set of numeric arrays of size d , denoted as \mathbb{R}^d . Here \mathbb{R} denotes the fact that components of each array are real numbers, and d denotes the number of components in each array. Thus, each element in \mathbb{R}^d is represented as $[x_1, x_2, \dots, x_d]$. Addition and scalar multiplication are element-wise as above, and the zero vector is the vector of all zeros.

What about the set of two dimensional numeric arrays? A two dimensional array, or a matrix, has rows and columns. An $n \times m$ matrix has n rows and m columns. If we consider the set of all $n \times m$ matrices, then we can denote this set as $\mathbb{R}^{n \times m}$ as before. Again, we can define addition and scalar multiplication element-wise. Convince yourself that this is indeed a vector space. Observe that if we consider *gray-scale images* of size $n \times m$ it is indeed exactly this vector space.

2 Bases and dimensionality

The point (x, y) on the cartesian plane can be thought of as “ x units along the X axis and y units along the Y axis”. Or, more precisely, $(x, y) = x(1, 0) + y(0, 1)$. In other words, any vector \mathbf{u} in the cartesian plane \mathbb{R}^2 can be represented as a *linear combination* of only two vectors, $(1, 0)$ and $(0, 1)$.

Similarly, consider the vector space \mathbb{R}^d . Consider the vectors $\mathbf{e}_1 = [1, 0, 0, \dots, 0]$, $\mathbf{e}_2 = [0, 1, 0, \dots, 0]$, $\mathbf{e}_3 = [0, 0, 1, \dots, 0]$ and so on. Thus \mathbf{e}_i has a 1 in the i -th position and is 0 everywhere else. Now consider

any vector $\mathbf{u} = [x_1, x_2, \dots, x_d]$. Then $\mathbf{u} = \sum_i x_i \mathbf{e}_i$. Again, any vector in \mathbb{R}^d can be represented as a linear combination of the \mathbf{e}_i 's.

What about the vector space of all $n \times m$ images, $\mathbb{R}^{n \times m}$? Recall that every element in this vector space is an $n \times m$ matrix. Consider the matrix \mathbf{e}_{ij} , which has a 1 in the (i, j) -th position and is zero everywhere

else. Then any vector $\mathbf{u} = \begin{bmatrix} x_{11} & x_{12} & \dots & x_{1m} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \dots & x_{nm} \end{bmatrix}$ in $\mathbb{R}^{n \times m}$ can be written as $\sum_{i,j} x_{ij} \mathbf{e}_{ij}$.

Thus it seems that the set of vectors $B_2 = \{(1, 0), (0, 1)\}$ in \mathbb{R}^2 , the set $B_d = \{\mathbf{e}_i; i = 1, \dots, d\}$ in \mathbb{R}^d and the set $B_{n \times m} = \{\mathbf{e}_{ij}; i = 1, \dots, n; j = 1, \dots, m\}$ in $\mathbb{R}^{n \times m}$ are all special in some way. Let us concretize this further.

We first need two definitions.

Definition 2 Let V be a vector space, and suppose $U \subset V$. Then a vector $\mathbf{v} \in V$ is said to be in the **span** of U if it is a linear combination of the vectors in U , that is, if $\mathbf{v} = \sum_{i=1}^n \alpha_i \mathbf{u}_i$ for some $\mathbf{u}_i \in U$ and some scalars α_i . The **span** of U is the set of all such vectors \mathbf{v} which can be expressed as linear combinations of vectors in U .

Thus, in \mathbb{R}^2 , the span of the set $B_2 = (0, 1), (1, 0)$ is all of \mathbb{R}^2 , since every vector in \mathbb{R}^2 can be expressed as a linear combination of vectors in B .

Definition 3 Let V be a vector space. A set of vectors $U = \mathbf{u}_1, \dots, \mathbf{u}_n \subset V$ is **linearly dependent** if there exist scalars $\alpha_1, \dots, \alpha_n$, not all of them 0, such that $\sum_i \alpha_i \mathbf{u}_i = \mathbf{0}$. If no such α_i 's exist, then the set of vectors U is **linearly independent**.

Consider, for example the set $(1, 0), (0, 1), (1, -1)$. Then, because $(1, 0) - (0, 1) - (1, -1) = \mathbf{0}$, this set is in fact linearly dependent. An equivalent definition for a **linearly independent set** is that no vector in the set is a linear combination of the others. This is because if \mathbf{u}_1 is a linear combination of $\mathbf{u}_2, \dots, \mathbf{u}_n$, then:

$$\mathbf{u}_1 = \sum_{i=2}^n \alpha_i \mathbf{u}_i \tag{1}$$

$$\Rightarrow \mathbf{0} = \sum_{i=2}^n \alpha_i \mathbf{u}_i - \mathbf{u}_1 \tag{2}$$

$$\Rightarrow \mathbf{0} = \sum_{i=1}^n \alpha_i \mathbf{u}_i \quad \text{where } \alpha_1 = -1 \tag{3}$$

$$\tag{4}$$

which in turn implies that $U = \{\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_n\}$ is linearly dependent.

Note that the sets B_2, B_d and $B_{n \times m}$ above are all linearly independent. Let us prove this via contradiction for B_d . Suppose that B_d is in fact linearly dependent. Then, there exist $\alpha_1, \dots, \alpha_d$, not all 0, s.t

$$\sum_{i=1}^d \alpha_i \mathbf{e}_i = \mathbf{0} \tag{5}$$

$$\Rightarrow [\alpha_1, \alpha_2, \dots, \alpha_d] = \mathbf{0} \tag{6}$$

$$\Rightarrow \alpha_i = 0 \forall i \tag{7}$$

which contradicts our assumption.

Definition 4 Let V be a vector space. A set of vectors $U \subset V$ is a **basis** for V if:

1. The span of U is V , that is, every vector in V can be written as a linear combination of vectors from U , and
2. U is linearly independent.

Thus B_2 is a basis for \mathbb{R}^2 , B_d is a basis for \mathbb{R}^d and $B_{n \times m}$ is a basis for $\mathbb{R}^{n \times m}$. However, note that a given vector space can have more than a single basis. For example, $B'_2 = (0, 1), (1, 1)$ is also a basis for \mathbb{R}^2 : any vector $(x, y) = x(1, 1) + (y - x)(0, 1)$, and it can be shown that $(1, 1)$ and $(0, 1)$ are linearly independent.

However, here is a crucial fact: all basis sets for a vector space *have the same number of elements*

Definition 5 *The number of elements in a basis for a vector space is called the **dimensionality** of the vector space.*

Thus the dimensionality of \mathbb{R}^2 is 2, the dimensionality of \mathbb{R}^d is d and the dimensionality of $\mathbb{R}^{n \times m}$ is nm . In general, the dimensionality of vector spaces can be infinite, but in computer vision we will only encounter finite-dimensional vector spaces.

2.1 Coordinate representation for vectors

A basis gives us a way to *represent* vectors in a unified way independent of the vector space. Let V be a vector space. Fix a basis $B = \mathbf{b}_1, \dots, \mathbf{b}_n$ for V . Then, for any vector \mathbf{x} in V , it can be represented as

$\sum_{i=1}^n \alpha_i \mathbf{b}_i$ for some α_i . We represent this vector as a column of numbers: $\begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_n \end{bmatrix}$.

Note that such a representation can be used only when we have chosen a basis. For some vector spaces such as \mathbb{R}^d , if no basis is mentioned, we will use B_d ; this is called the canonical basis.

3 Norms, distances and angles

For many vector spaces, we also want to have a notion of lengths and distances, and similarities between two vectors. For example, we may want to ask, how far is the point $(4, 5)$ from the point $(0, 0)$? In the cartesian plane, there is a natural notion of distance. For example, to get to $(4, 5)$ from $(0, 0)$, we will have to go 4 unit along X and 5 units along Y; then we can use Pythagoras theorem to compute the distance.

The generalization of this computation is a norm. Although there are many different norms for many vector spaces, for our purposes in computer vision, we mainly deal with the L_2 norm for \mathbb{R}^d , which is denoted by $\|\cdot\|_2$ or $\|\cdot\|$. This norm is defined as follows: $\|\mathbf{x}\| = \sqrt{x_1^2 + x_2^2 + \dots + x_d^2}$

The norm of a vector \mathbf{x} is its *length*. The distance between two vectors \mathbf{x} and \mathbf{y} is the length of $\mathbf{x} - \mathbf{y}$. Often, we will talk of the *direction* of a vector \mathbf{x} , which is just \mathbf{x} multiplied by a scalar to make its norm 1: $\frac{\mathbf{x}}{\|\mathbf{x}\|}$. Such a vector is also called a *unit vector*.

Another useful quantity is the dot product or inner product between two vectors, denoted as $\langle \cdot, \cdot \rangle$. Again, there are many possible inner products we can use. However, the most common inner product for \mathbb{R}^d is given by $\langle \mathbf{x}, \mathbf{y} \rangle = x_1 y_1 + x_2 y_2 + \dots + x_d y_d$. Observe that, with this definition of inner product, $\langle \mathbf{x}, \mathbf{x} \rangle = \|\mathbf{x}\|^2$.

This inner product generalizes the dot product you may have encountered in 3D geometry. It can be shown that with this definition, $\langle \mathbf{x}, \mathbf{y} \rangle = \|\mathbf{x}\| \|\mathbf{y}\| \cos \theta$, where θ is the angle between \mathbf{x} and \mathbf{y} . Thus an inner product of 0 between two non-zero vectors indicates that they are perpendicular (or *orthogonal*) to each other. This connection with the angle also motivates the use of this inner product as a measure of similarity between two vectors.

As a matter of notation, note that vectors written as column vectors can also be thought of as $d \times 1$ matrices. As such, the inner product described above is also equivalent to $\mathbf{x}^T \mathbf{y}$, and is commonly represented in this way.

4 Linear transformations

Suppose we have two vector spaces U and V . We are interested in functions that map from one to the other. Perhaps the most important class of these functions in terms of their practical uses as well as ease of understanding is the class of *linear transformations*.

Definition 6 Consider two vector spaces U and V . A function $f : U \rightarrow V$ is a linear transformation if $f(\alpha \mathbf{u}_1 + \beta \mathbf{u}_2) = \alpha f(\mathbf{u}_1) + \beta f(\mathbf{u}_2)$ for all scalars α, β and for all $\mathbf{u}_1, \mathbf{u}_2 \in U$.

Let $f : U \rightarrow V$ be a linear transformation. Suppose that we have fixed a basis $B_U = \mathbf{b}_1, \dots, \mathbf{b}_m$ for U , and a basis $B_V = \mathbf{a}_1, \dots, \mathbf{a}_n$ for V . Consider the vectors $f(\mathbf{b}_j), j = 1, \dots, m$. Since these are vectors in V , they can be expressed as a linear combination of vectors in B_V . Thus:

$$f(\mathbf{b}_j) = \sum_{i=1}^n M_{ij} \mathbf{a}_i \quad (8)$$

for some coefficients M_{ij} .

Now consider an arbitrary vector \mathbf{u} in U . It should be expressible as a linear combination of the basis vectors, so $\mathbf{u} = \sum_{j=1}^m u_j \mathbf{b}_j$. We can now figure out what $f(\mathbf{u})$ should be:

$$f(\mathbf{u}) = f\left(\sum_{j=1}^m u_j \mathbf{b}_j\right) \quad (9)$$

$$= \sum_{j=1}^m u_j f(\mathbf{b}_j) \quad (\text{by linearity of } f) \quad (10)$$

$$= \sum_{j=1}^m \sum_{i=1}^n M_{ij} u_j \mathbf{a}_i \quad (11)$$

Now we can express \mathbf{u} as a column vector of coefficients $\begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_m \end{bmatrix}$. If we express $f(\mathbf{u})$ as a column vector

similarly, we can see that:

$$f(\mathbf{u}) = M\mathbf{u} \quad (12)$$

where $M = \begin{bmatrix} M_{11} & \dots & M_{1m} \\ \vdots & \ddots & \vdots \\ M_{n1} & \dots & M_{nm} \end{bmatrix}$.

Thus **every linear transformation can be expressed as a matrix multiplication**. The matrix encodes *how each basis vector gets transformed*; the linearity of the transformation means that this information is enough to predict how everything else will be transformed. In particular, the j -th column of this matrix is the transformed j -th basis vector (Equation (8)). You should also be able to prove that **every matrix multiplication is a linear transformation**.

4.1 Change of basis

A special case of a linear transformation is when we want to change the basis. In this case, U and V above are the same vector space, but with different basis. Given a vector \mathbf{u} represented in the basis B_U , we want to know its representation in the basis B_V . Again, this can be represented using a matrix multiplication $M\mathbf{u}$. Here, the j -th column of M is the representation of the j -th basis vector in B_U , namely \mathbf{b}_j , now represented in the basis B_V .

4.2 Rank and nullity

Consider a linear transformation from vector space U to V corresponding to the matrix M . As can be seen

above, the output of this transformation is a linear combination of the matrix column vectors $\mathbf{m}_j = \begin{bmatrix} M_{1j} \\ \vdots \\ M_{nj} \end{bmatrix}$

Thus, the output is the *span* of the matrix columns. Note that this span need not be the whole space V . It can be a *subset* of V , but *still a vector space* (it should be easy to verify that the set of all linear combinations of a set of vectors is itself a vector space, a subset of the original space; often called a *subspace*). The dimensionality of this output subspace is called the **rank** of the matrix M . If this dimensionality is equal to the dimensionality of V , the matrix M is considered *full rank*.

As another useful property, consider the set of vectors \mathbf{u} in U s.t $M\mathbf{u} = \mathbf{0}$. Again, it can be shown that this set is a vector space, and is thus a *subspace* of U . The dimensionality of this subspace is called the **nullity** of the matrix M .

One of the most useful theorems of linear algebra is that **rank + nullity = number of columns of M** .

4.3 Properties of matrices

Because matrices represent linear transformations between vector spaces, a lot of linear algebra is devoted to studying them. For our purposes, there are several properties of matrices that you must know :

1. The matrix I is such that $I_{ii} = 1$ for all i , and $I_{ij} = 0$ if $j \neq i$. This is the identity matrix and represents an identity transformation: $I\mathbf{x} = \mathbf{x}$.
2. Matrices can be multiplied together as follows. If $C = AB$, then $C_{ij} = \sum_k A_{ik}B_{kj}$. Note that this means that A and B can only be multiplied together if the number of columns of A match the number of rows of B . The product of an $n \times m$ matrix and an $m \times l$ matrix is $n \times l$.
3. Matrix multiplication is associative ($A(BC) = (AB)C$) and distributes over addition ($A(B + C) = AB + AC$) but it is *not commutative in general* (i.e., AB need not equal BA)
4. For a matrix A , we can construct another matrix B s.t $B_{ij} = A_{ji}$. B is called the transpose of A and is denoted as A^T .
5. If $A = A^T$, A is *symmetric*. If $A = -A^T$, A is *skew-symmetric*.
6. For a matrix A , if B is such that $AB = I$, B is a *right-inverse* of A . If $BA = I$, B is a *left-inverse* of A . If $BA = AB = I$, B is an inverse of A . Inverses when they exist are unique and are denoted as A^{-1} .
7. Square matrices A such that $AA^T = A^T A = I$ are called *orthonormal* matrices. Note that for any orthonormal matrix R , if \mathbf{r}_j is the j -th column of R , then $\mathbf{r}_j^T \mathbf{r}_j = 1$ and $\mathbf{r}_j^T \mathbf{r}_i = 0$ for $i \neq j$. Thus, columns of an orthonormal matrix have unit norm and are orthogonal to each other.
8. The *trace* of a matrix is the sum of its diagonal elements (i.e., $trace(A) = \sum_i A_{ii}$)
9. Another useful function of a *square* matrix is the *determinant*, denoted as $det(A)$. The determinant can be difficult to compute. For a 2×2 matrix $\begin{bmatrix} a & b \\ c & d \end{bmatrix}$, the determinant is $ad - bc$. For a diagonal matrix it is the product of the eigenvalues. For matrices that are not full rank, the determinant is 0. The determinant of $AB = det(A)det(B)$.

5 Understanding and decomposing linear transformations

Because matrices and linear transforms are so important, it is worthwhile delving into them a bit more. In particular, let us focus on linear transformations $f : \mathbb{R}^n \rightarrow \mathbb{R}^n$, which correspond to $n \times n$ matrices M . Consider an input vector \mathbf{x} , and the corresponding output vector $M\mathbf{x}$. We want to understand how $M\mathbf{x}$ relates to \mathbf{x} , and what that means in terms of properties of M .

Table 1 shows three kinds of 2×2 matrices and their effect on a set of 2D points spread evenly on the unit circle:

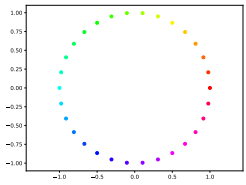
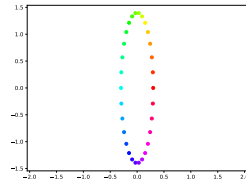
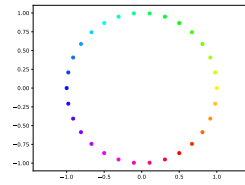
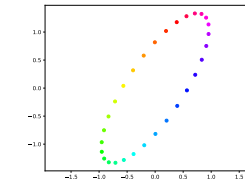
	Scaling	Rotation	General transformation
			
Input points	$\begin{bmatrix} 0.3 & 0 \\ 0 & 1.4 \end{bmatrix}$	$\begin{bmatrix} 0.5 & 0.866 \\ -0.866 & 0.5 \end{bmatrix}$	$\begin{bmatrix} 0.38 & -0.88 \\ 1.18 & -0.63 \end{bmatrix}$

Table 1: Three different matrices and their action on a set of points.

- Column 2 shows a *scaling* (non-isometric), which involves stretching or compressing the axes to different

extents. Matrices that perform such operations are *diagonal matrices* of the form $\begin{bmatrix} s_1 & 0 & \dots & 0 \\ 0 & s_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & s_n \end{bmatrix}$.

- Column 3 shows a *rotation* which, as the name suggests, involves a rotation about the origin. Rotation matrices R are *orthonormal* ($R^T R = I$) and in addition have determinant 1. A close cousin are *reflections* which also involve orthonormal matrices but with determinant -1 .

- Column 4 shows a general linear transformation.

It can be seen that the transformation in Column 4 involves some scaling along an arbitrary axis as well as some rotation. It turns out that a version of this holds for *any* linear transformation. In particular, every linear transformation can be decomposed into a rotation/reflection, followed by a non-isometric scaling, followed by another rotation/reflection. This decomposition is called a singular value decomposition:

Definition 7 Every matrix M can be written as $M = U\Sigma V^T$, where U and V are orthonormal matrices and Σ is a diagonal matrix. This is known as the **singular value decomposition (SVD)** of matrix M . The values in the diagonal of Σ are called the singular values of the matrix M .

Thus $M\mathbf{x} = U(\Sigma(V^T\mathbf{x}))$. In other words, applying M amounts to rotating using V^T , scaling using Σ and rotating again using U . For any matrix M , Σ is unique, and U and V are unique upto a sign flip for each column.

Singular value decomposition is one of the most *versatile, commonly used tools of linear algebra* and it is worthwhile remembering that this exists. Apart from being an interpretable decomposition of a matrix, it offers several interesting properties.

- The *rank* of the matrix M is simply the number of non-zero singular values.
- Given a matrix $M = U\Sigma V^T$ of rank r , if we want the closest matrix M' of rank $r - k$, then one can simply zero out the k *smallest singular values* (smallest in absolute value) in Σ to produce Σ' . M' is then $U\Sigma'V^T$.
- If \mathbf{u}_i is the i -th column of U , \mathbf{v}_i is the i -th column of V and σ_i is the i -th diagonal element of Σ , then it is easy to show that $M = \sum_i \sigma_i \mathbf{u}_i \mathbf{v}_i^T$.
- Consider what happens when M is applied to the vector \mathbf{v}_j . Since $\mathbf{v}_i^T \mathbf{v}_j = \begin{cases} 0 & \text{if } i \neq j \\ 1 & \text{ow} \end{cases}$, we have:

$$M\mathbf{v}_j = \sum_i \sigma_i \mathbf{u}_i \mathbf{v}_i^T \mathbf{v}_j = \sigma_j \mathbf{u}_j \quad (13)$$

Consider now square, symmetric matrices, that is $M = M^T$. If $M = U\Sigma V^T$, then this means that $U\Sigma V^T = V\Sigma U^T$, or in other words $U = V$. In this case the singular value decomposition coincides with another matrix decomposition, the eigenvalue decomposition:

Definition 8 Every square matrix M can be written as $M = U\Lambda U^T$, where U is an orthonormal matrix and Λ is a diagonal matrix. This is known as the **eigenvalue decomposition** of matrix M . The values in the diagonal of Λ are called the eigenvalues of the matrix M .

As above, if \mathbf{u}_j is the j -th column of U (or the j -th *eigenvector*) and λ_j is the j -th eigenvalue, then:

$$M\mathbf{u}_j = \lambda_j\mathbf{u}_j \tag{14}$$

Thus, eigenvectors of M are vectors which when multiplied by M will point in the same direction, but have their norm scaled by λ .

All square matrices have an eigenvalue decomposition, but the eigenvalues and eigenvectors may be complex. If the matrix is symmetric, then the eigenvectors and eigenvalues will be real and the eigenvalue decomposition coincides with the SVD.

An interesting fact is that the eigenvectors of a square $d \times d$ matrix always form a *basis* for \mathbb{R}^d . Similarly, the column vectors of V in an SVD of a $d \times d$ matrix form a basis for \mathbb{R}^d .

6 Matrices, vectors and optimization

One of the primary use-cases of linear algebra techniques will appear when we try to solve equations or optimization problems.

6.1 Solving linear equations

Consider, for example, any set of linear equations in 2 variables:

$$a_{11}x + a_{12}y = b_1 \tag{15}$$

$$a_{21}x + a_{22}y = b_2 \tag{16}$$

$$\vdots \tag{17}$$

$$a_{n1}x + a_{n2}y = b_n \tag{18}$$

This set of equations can be written using matrices and vectors, where we assemble all the unknowns into a single vector $\mathbf{x} = \begin{bmatrix} x \\ y \end{bmatrix}$:

$$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \\ \vdots & \vdots \\ a_{n1} & a_{n2} \end{bmatrix} \mathbf{x} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix} \tag{19}$$

$$\Rightarrow A\mathbf{x} = \mathbf{b} \tag{20}$$

In fact, any set of linear equations in d variables can be written as a matrix vector equation $A\mathbf{x} = \mathbf{b}$, where A and \mathbf{b} are a vector of coefficients and \mathbf{x} is a vector of the variables. In general, if A is $d \times d$ and full rank, then A^{-1} exists and the solution to these equations are simply $\mathbf{x} = A^{-1}\mathbf{b}$. However, what if A is not full rank or different from $d \times d$?

Over-constrained systems What happens if A is $n \times d$, where $n > d$, and is full rank? In this case, there are more constraints than there are variables, so there may not in fact be a solution. Instead, we look for a solution in the least squares sense: we try to optimize:

$$\min_{\mathbf{x}} \|A\mathbf{x} - \mathbf{b}\|^2 \tag{21}$$

Now, we have:

$$\|A\mathbf{x} - \mathbf{b}\|^2 = (A\mathbf{x} - \mathbf{b})^T(A\mathbf{x} - \mathbf{b}) \quad (22)$$

$$= (\mathbf{x}^T A^T - \mathbf{b}^T)(A\mathbf{x} - \mathbf{b}) \quad (23)$$

$$= \mathbf{x}^T A^T A\mathbf{x} - \mathbf{b}^T A\mathbf{x} - \mathbf{x}^T A^T \mathbf{b} + \mathbf{b}^T \mathbf{b} \quad (24)$$

$$= \mathbf{x}^T A^T A\mathbf{x} - 2\mathbf{b}^T A\mathbf{x} + \mathbf{b}^T \mathbf{b} \quad (25)$$

We now must minimize this function over \mathbf{x} . This can be done by computing the derivative of this objective w.r.t each component of \mathbf{x} and setting it to 0. In vector notation, the vector of derivatives of a function $f(\mathbf{x})$ with respect to each component of \mathbf{x} is called the *gradient* $\nabla_{\mathbf{x}} f(\mathbf{x})$:

$$\nabla_{\mathbf{x}} f(\mathbf{x}) = \begin{bmatrix} \frac{\partial f(\mathbf{x})}{\partial x_1} \\ \frac{\partial f(\mathbf{x})}{\partial x_2} \\ \vdots \\ \frac{\partial f(\mathbf{x})}{\partial x_d} \end{bmatrix} \quad (26)$$

We will use two identities here that are easy to prove:

$$\nabla_{\mathbf{x}} \mathbf{c}^T \mathbf{x} = \mathbf{c} \quad (27)$$

$$\nabla_{\mathbf{x}} \mathbf{x}^T Q \mathbf{x} = (Q + Q^T) \mathbf{x} \quad (28)$$

This gives us:

$$\nabla_{\mathbf{x}} (\mathbf{x}^T A^T A\mathbf{x} - 2\mathbf{b}^T A\mathbf{x} + \mathbf{b}^T \mathbf{b}) = 2A^T A\mathbf{x} - 2A^T \mathbf{b} \quad (29)$$

Setting this to 0 gives us the *normal equations*, which are now precisely a set of d equations:

$$A^T A\mathbf{x} = A^T \mathbf{b} \quad (30)$$

These can be solved the usual way giving us the least squares solution $\mathbf{x} = (A^T A)^{-1} A^T \mathbf{b}$.

Under-constrained equations What if A has rank $n < d$? In this case, there might be multiple possible solutions, and the system is *underconstrained*. It is also possible that no solution exists. In particular, if \mathbf{x}_1 is a solution (i.e., $A\mathbf{x}_1 = \mathbf{b}$), and $A\mathbf{x}_2 = \mathbf{0}$, then $\mathbf{x}_1 + \mathbf{x}_2$ is also a solution.

We can get a *particular* solution as follows. First we do an SVD of A to get:

$$U\Sigma V^T \mathbf{x} = \mathbf{b} \quad (31)$$

$$\Leftrightarrow \Sigma V^T \mathbf{x} = U^T \mathbf{b} \quad (32)$$

$$(33)$$

Next, let $\mathbf{y} = V^T \mathbf{x}$, so that $\mathbf{x} = V\mathbf{y}$. Then,

$$\Sigma \mathbf{y} = U^T \mathbf{b} \quad (34)$$

Because Σ is a diagonal matrix, this equation can be solved trivially, if a solution exists (note that since A is not full rank, some diagonal entries of Σ are 0; the corresponding entries of the RHS must be 0 for a solution to exist).

6.2 Optimization problems

Another common use-case of linear algebra is for solving optimization problems. Consider the problem: $\min_{\mathbf{x}} \mathbf{x}^T Q \mathbf{x}$, subject to the constraint that $\|\mathbf{x}\| = 1$. Here Q is symmetric. To solve this, let us express \mathbf{x} as

a linear combination of the eigenvectors of Q : $\mathbf{x} = \sum_i \alpha_i \mathbf{v}_i$. Then, we have

$$\mathbf{x}^T Q \mathbf{x} = \left(\sum_i \alpha_i \mathbf{v}_i^T \right) Q \left(\sum_j \alpha_j \mathbf{v}_j \right) \quad (35)$$

$$= \left(\sum_i \alpha_i \mathbf{v}_i^T \right) \left(\sum_j \alpha_j Q \mathbf{v}_j \right) \quad (36)$$

$$= \left(\sum_i \alpha_i \mathbf{v}_i^T \right) \left(\sum_j \alpha_j \lambda_j \mathbf{v}_j \right) \quad (37)$$

$$= \sum_{i,j} \alpha_i \alpha_j \lambda_j \mathbf{v}_i^T \mathbf{v}_j \quad (38)$$

$$= \sum_i \alpha_i^2 \lambda_i \quad (39)$$

Thus, the objective function is a linear combination of the λ_j with positive weights α_j^2 . The only way to minimize this is to put maximum weight α_j on the *smallest eigenvalue* and 0 weight on everything else. The maximum weight we can put is 1, since $\|\mathbf{x}\| = 1$. Thus the solution to the minimization is \mathbf{v}^* , the eigenvector corresponding to the smallest eigenvalue λ_* .