Linear algebra with applications

Ivan Markovsky
ELEC, Vrije Universiteit Brussel
`ivan.markovsky@vub.ac.be`

Contents

Chapter 1

Review of linear algebra

- Linear functions and matrix–vector product
- Rank of a matrix and inversion
- Inner product
- Subspaces, basis, and dimension
- Eigenvalues and eigenvectors

1.1 Linear functions and matrix–vector product

Linear functions

Standard notation for a function f mapping a vector $x \in \mathbb{R}^n$ to a vector $y \in \mathbb{R}^m$ is

$$f : \mathbb{R}^n \to \mathbb{R}^m \qquad \text{or} \qquad f : x \mapsto y.$$

The value $y \in \mathbb{R}^m$ of f at $x \in \mathbb{R}^n$ is denoted by $y = f(x)$. Note that f and $f(x)$ are different objects—f is a function and $f(x)$ is a vector. Therefore, the statement "the function $f(x)$" is semantically wrong, despite the fact that its meaning is intuitively clear and is commonly used.

A function f is usually specified by an analytic expression, *e.g.*, $f(x) = x^2$, but it can be specified in other ways as well. For example, a function f can be defined by an algorithm that evaluates f for a given x or by a verbal description, *e.g.*, "$f(x)$ is the vector x rotated clockwise by α°". In system theory a function f is visualized by a box, called a *system*, that accepts as an *input* x and produces as an *output* y.

$$\begin{tikzcd} x \arrow[r] & \boxed{f} \arrow[r] & y \end{tikzcd}$$

One can think of the system as a device or a signal processor that transforms energy or information. However, a system in system theory is an abstract object and is distinguished from a physical device.

By definition, f is a *linear function* if the following property holds:

$$f(\alpha x + \beta z) = \alpha f(x) + \beta f(z), \quad \text{for all } \alpha, \beta \in \mathbb{R} \text{ and } x, z \in \mathbb{R}^n.$$

An equivalent definition is that f satisfies the *homogeneity* and *superposition* properties

- homogeneity: $f(\alpha x) = \alpha f(x)$, for all $\alpha \in \mathbb{R}$, and $x \in \mathbb{R}^n$,
- superposition: $f(x + z) = f(x) + f(z)$, for all $x, z \in \mathbb{R}^n$.

Exercise problem 1. Show that rotation is a linear function.

□

Matrix–vector product

Partition a matrix $A \in \mathbb{R}^{m\times n}$ elementwise, column-wise, and row-wise, as follows

$$A = \begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & & \vdots \\ a_{m1} & \cdots & a_{mn} \end{bmatrix} = \begin{bmatrix} | & & | \\ a_1 & \cdots & a_n \\ | & & | \end{bmatrix} = \begin{bmatrix} — & (a'_1)^\top & — \\ & \vdots & \\ — & (a'_m)^\top & — \end{bmatrix}.$$

The matrix–vector product $y = Ax$ can be written in three alternative ways corresponding to the three partitionings above

$$\begin{bmatrix} y_1 \\ \vdots \\ y_m \end{bmatrix} = \begin{bmatrix} \sum_{j=1}^n a_{1j}x_j \\ \vdots \\ \sum_{j=1}^n a_{mj}x_j \end{bmatrix} = \sum_{j=1}^n a_j x_j = \begin{bmatrix} (a'_1)^\top x \\ \vdots \\ (a'_m)^\top x \end{bmatrix}.$$

For a given A, $y = Ax$ defines a function $f : x \mapsto y$. Matrix-vector product, however, is more than an example of a linear function. It is the only example in the sense that any linear function admits a representation in the form of a matrix times vector.

Exercise problem 2. Prove that the function $f : \mathbb{R}^n \to \mathbb{R}^m$ is linear if and only if there is a matrix $A \in \mathbb{R}^{m\times n}$, such that $f(x) = Ax$, for all $x \in \mathbb{R}^n$.

□

The matrix A is called a matrix representation of the function f, $f(x) = Ax$. Given a matrix representation A of a linear function f, the problem of evaluating the function $y = f(x)$ at a given point x is a matrix–vector multiplication $y = Ax$ problem.

Note 3. Formally one should make a distinction between a vector and a vector representation. A vector representation depends on the choice of basis and is therefore not unique. Similarly, a matrix representation of a linear function depends on the bases of the input space $\mathbb{R}^n$ and the output space $\mathbb{R}^m$ and is not unique, see (1.7) in Section 1.4.

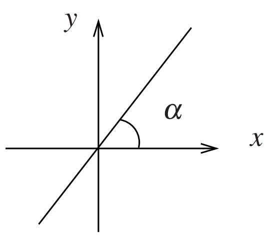
Exercise problem 4. Explain how to find a matrix representation of a linear function f, if f can be evaluated at arbitrary points $x \in \mathbb{R}^n$. Apply the procedure to the rotation function in $\mathbb{R}^n$.

□

Example 5. A scalar linear function of a scalar argument

$$y = \tan(\alpha)x, \qquad \text{where} \qquad \alpha \in [0, 2\pi)$$

is a line in the plain passing through the origin. Its matrix representation is the scalar $\tan(\alpha)$. Conversely, any line in the plain passing though the origin is a linear function.



Example 6. A scalar valued linear function of a vector argument $f : \mathbb{R}^n \to \mathbb{R}$ is given by $f(x) = a^\top x$, where $a \in \mathbb{R}^n$. (The expression $a^\top x$, *i.e.*, row vector times column vector, is called inner product, see Section 1.3.)

Example 7. The *identity function* $x = f(x)$, for all $x \in \mathbb{R}^n$, is a linear function represented by the $n\times n$ identity matrix

$$I_n := \begin{bmatrix} 1 & & \\ & \ddots & \\ & & 1 \end{bmatrix} \in \mathbb{R}^{n\times n}.$$

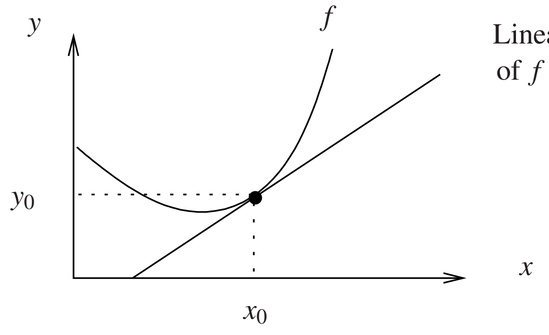
Consider a differentiable function $f : \mathbb{R}^n \to \mathbb{R}^m$. Then for a given $x_0 \in \mathbb{R}^n$

$$y = f(x_0 + \widetilde{x}) \approx f(x_0) + A\widetilde{x}, \qquad \text{where} \quad A = \left[a_{ij}\right] = \left[\left. \frac{\partial f_i}{\partial x_j} \right|_{x_0} \right].$$

($\partial f_i / \partial x_j$ is the partial derivative of f_i with respect to x_j and the matrix A of the partial derivatives is called the Jacobian of f.) When the input deviation $\widetilde{x} = x - x_0$ is “small”, the output deviation

$$\widetilde{y} := y - f(x_0) =: y - y_0$$

is approximately the linear function $\widetilde{y} = A\widetilde{x}$ of $\widetilde{x}$, called the linear approximation of f at x_0.



Linear approximation of $f : x \mapsto y$ at $x = x_0$

1.2 Rank of a matrix and inversion

The set of vectors $\{a_1,\dots,a_n\}$ is *linearly independent* if the only linear combination of these vectors that is equal to the zero vector is the trivial linear combination with all weights equal to zero, *i.e.*,

$$x_1a_1+\dots+x_na_n=0 \qquad \Longrightarrow \qquad x_1=\dots=x_n=0.$$

Linear independence means that non of the vectors a_i, for $i=1,\dots,n$, can be expressed as a linear combination of the remaining vectors. Vice verse, in a linearly dependent set of vectors $\{a_1,\dots,a_n\}$ at least one vector is equal to a linear combination of the others. This means that in a linearly dependent set of vectors, there is redundant information.

The rank $\operatorname{rank}(A)$ of the matrix $A\in\mathbb{R}^{m\times n}$ is the number of linearly independent columns (or rows) of A and zero if A is the zero matrix. Obviously,

$$0\le\operatorname{rank}(A)\le\min(m,n)$$

The matrix A is

- *full row rank* if $\operatorname{rank}(A)=m$,
- *full column rank* if $\operatorname{rank}(A)=n$, and
- *full rank* if it is either full row rank or full column rank.

Full row and column rank of A turns out to be a necessary and sufficient condition for, respectively, existence of solution of the system $Ax=y$, for *any* given $y\in\mathbb{R}^m$, and uniqueness of a solution of $Ax=y$ for any given $y\in\mathbb{R}^m$. (Existence of solution of $Ax=y$, does depend on both A and y. However, assuming that the system $Ax=y$ is solvable, the uniqueness of a solution x depends only on A.)

Exercise problem 8. Prove that the matrix A being full row rank is equivalent to the system of equations $Ax=y$ having a solution for *any* $y\in\mathbb{R}^m$.

Exercise problem 9. Prove that the matrix A being full column rank is equivalent to uniqueness of a solution x to the system $y=Ax$, where $y=A\bar{x}$ for some $\bar{x}\in\mathbb{R}^n$.

A system theoretic interpretation of A being full row rank is that the system defined by $y=Ax$ has no redundant outputs, *i.e.*, none of the components of y can be inferred from the others. An interpretation of A being full column rank is that the there exists an inverse system, *i.e.*, a mechanism (which is also a system) to infer the the input from the output.

Next we consider the inversion problem: given $y\in\mathbb{R}^m$, find x, such that $y=Ax$. We distinguish three cases depending on the shape of the matrix A (square, more rows than columns, or more columns than rows) and in all cases we assume that A is full rank.

- If $m=n=\operatorname{rank}(A)$, then there exists a matrix A^{-1}, such that

  $$AA^{-1}=A^{-1}A=I_m. \tag{1.1}$$

  Then for all $y\in\mathbb{R}^m$

  $$y=\underbrace{(AA^{-1})}_{I_m}y=A\underbrace{(A^{-1}y)}_{x}=Ax.$$

  In this case, the inversion problem is solvable and the solution is unique.

  Exercise problem 10. Prove the fact that $m=n=\operatorname{rank}(A)$ implies existance of a matrix A^{-1}, such that (1.1).

  Exercise problem 11. Find a matrix representation of a linear function f, from given values $y_1,\dots,y_n$ of f at given points $x_1,\dots,x_n$. When is this problem solvable?

$\square$

- If $m \geq n = \operatorname{rank}(A)$, *i.e.*, A is full column rank, the inversion problem may have no solution. In such cases, an approximate solution may be desirable. The least-squares approximate solution minimizes the 2-norm

  $$\|e\|_2 := \sqrt{e^\top e} = \sqrt{e_1^2 + \cdots + e_n^2}, \tag{1.2}$$

  of the *approximation error* (or *residual*)

  $$e := y - Ax.$$

  The least-squares approximation problem is

  $$\text{minimize } \|e\|_2 \quad \text{subject to} \quad Ax = y + e$$

  and the solution is given by the famous formula

  $$x_{\text{ls}} = (A^\top A)^{-1} A^\top y =: A_{\text{ls}}^{\text{L}} y. \tag{1.3}$$

  Note that x_{ls} is a linear function $A_{\text{ls}}^{\text{L}} y$ of y. It is called the least squares approximate solution of the system of equations $Ax = y$. If $y = A\bar{x}$, for some $\bar{x} \in \mathbb{R}^n$, x_{ls} is an exact solution, *i.e.*, $x_{\text{ls}} = \bar{x}$.

  Note 12 (Left inverse). Any matrix A^{L}, satisfying the property $A^{\text{L}} A = I_n$ is called a *left inverse* of A. Left inverse of A exists if and only if A is full column rank. If $m > n$, the left inverse is nonunique. If $m = n$, the left inverse is unique and is equal to the inverse. The matrix A_{ls}^{L} is a left inverse of A. Moreover, it is the smallest left inverse, in the sense that it minimizes the Frobenius norm

  $$\|A^{\text{L}}\|_{\text{F}} := \sqrt{\sum_{i=1}^{m} \sum_{j=1}^{n} (a_{ij}^{\text{L}})^2}$$

  over all left inverses A^{L} of A.

  Exercise problem 13. Prove that $A_{\text{ls}}^{\text{L}} = \arg\min_{A^{\text{L}}} \|A^{\text{L}}\|_{\text{F}}$ subject to $A^{\text{L}} A = I$.

  $\square$

- If $n \geq m = \operatorname{rank}(A)$, *i.e.*, A is full row rank, the inversion problem has infinitely many solutions. The set of all solutions is

  $$\{\, x \mid Ax = y \,\} = \{\, x_{\text{p}} + z \mid Az = 0 \,\}, \qquad \text{where} \quad Ax_{\text{p}} = y,$$

  i.e., x_{p} is a particular solution of $Ax = y$ and z is a parameter describing the nonuniquness of the solution. The *least-norm solution* is

  $$\text{minimize } \|x\|_2 \quad \text{subject to} \quad Ax = y.$$

  It is given by the following closed form expression

  $$x_{\text{ln}} = A^\top (AA^\top)^{-1} y =: A_{\text{ln}}^{\text{R}} y. \tag{1.4}$$

  Note 14 (Right inverse). Any matrix A^{R}, satisfying the property $AA^{\text{R}} = I_m$ is called a *right inverse* of A. Right inverse of A exists if and only if A is full row rank. If $m < n$, the right inverse is nonunique. If $m = n$, the right inverse is unique and is equal to the inverse. The matrix A_{ln}^{R} is a right inverse of A. Moreover, it can be shown that it is the smallest right inverse, in the sense that it minimizes the Frobenius norm $\|A^{\text{R}}\|_{\text{F}}$ over all right inverses A^{R} of A.

  Exercise problem 15. Prove that $A_{\text{ln}}^{\text{R}} = \arg\min_{A^{\text{R}}} \|A^{\text{R}}\|_{\text{F}}$ subject to $AA^{\text{R}} = I$.

  $\square$

Note 16 (Inversion problem in the singular case). If $A \in \mathbb{R}^{m \times n}$ is rank deficient (or almost rank deficient), the inversion problem is called *ill-posed* (or *ill-conditioned*). In this case, the inverse (assuming A is square) does not exist. Also the least-squares (1.3) (assuming $m > n$) and the least-norm (1.4) (assuming $m < n$) formulas make no sense (because the indicated inverses do not exist). A general solution to the inversion problem, which is independent of size and rank assumptions on A, is given by $\widehat{x} = A^{+} y$, what $A^{+} \in \mathbb{R}^{n \times m}$ is the *pseudo-inverse* of A. A related approach for solving ill-posed and ill-conditioned inverse problems is *regularization*.

1.3 Inner product

The inner product $\langle a,b\rangle = \langle b,a\rangle$ of two vectors $a,b \in \mathbb{R}^m$ is defined by

$$\langle a,b\rangle := a^\top b = \sum_{i=1}^{m} a_i b_i.$$

The matrix–matrix product BA, where $B : \mathbb{R}^{p\times m}$ and $A : \mathbb{R}^{m\times n}$, can be viewed as a collection of pn inner products (between the rows of B and the columns of A)

$$BA = \begin{bmatrix} — & (b'_1)^\top & — \\ & \vdots & \\ — & (b'_p)^\top & — \end{bmatrix} \begin{bmatrix} | & & | \\ a_1 & \cdots & a_n \\ | & & | \end{bmatrix} = \begin{bmatrix} \langle b'_1 a_1\rangle & \cdots & \langle b'_1, a_n\rangle \\ \vdots & & \vdots \\ \langle b'_p, a_1\rangle & \cdots & \langle b'_p, a_n\rangle \end{bmatrix}.$$

The *Gram matrix* of the vectors $a_1,\dots,a_n$ is defined by

$$\begin{bmatrix} a_1^\top \\ \vdots \\ a_n^\top \end{bmatrix} \begin{bmatrix} a_1 & \cdots & a_n \end{bmatrix} = A^\top A.$$

The Gram matrix $H := A^\top A$ is *symmetric*, *i.e.*, $H = H^\top$ and *positive semidefinite*, *i.e.*, $x^\top Hx \geq 0$, for all $x \in \mathbb{R}^n$. A matrix H is called *positive definite* if $x^\top Hx > 0$, for all $x \in \mathbb{R}^m$.

Exercise problem 17. The Gram matrix $A^\top A$ is positive definite if and only if A is full column rank.

□

The Cauchy-Schwarz inequality relates the inner product with the product of the 2-norms

$$\langle a,b\rangle \leq \|a\|\|b\|. \tag{1.5}$$

Equality holds in (1.5) if and only if $b = \alpha a$, for some $\alpha \in \mathbb{R}$ or $b = 0$.

Exercise problem 18. Prove (1.5).

□

Exercise problem 19 (Optimization of a linear function over the unit ball). Show that the solution of the problem, given $a \in \mathbb{R}^n$,

$$\text{maximize} \quad (\text{over } x) \quad a^\top x \quad \text{subject to} \quad \|x\| \leq 1$$

is $x_{\text{opt}} = a/\|a\|$.

□

The angle between the vectors $a,b \in \mathbb{R}^n$ is defined as

$$\angle(a,b) = \cos^{-1} \frac{a^\top b}{\|a\|\|b\|}.$$

- $a \neq 0$ and b are *aligned* if $b = \alpha a$, for some $\alpha \geq 0$ (in this case, $\angle(a,b) = 0$).
- $a \neq 0$ and b are *opposite* if $b = -\alpha a$, for some $\alpha \geq 0$ (in this case, $\angle(a,b) = \pi$).
- a and b are *orthogonal*, denoted $a \perp b$, if $a^\top b = 0$ (in this case, $\angle(a,b) = \pi/2$).

1.4 Subspace, basis, and dimension

The set $\mathscr{A} \subset \mathbb{R}^n$ is a *subspace* of a vector space $\mathbb{R}^n$ if $\mathscr{A}$ is a vector space itself, *i.e.*,

$$a,b \in \mathscr{A} \quad \Longrightarrow \quad \alpha a + \beta b \in \mathscr{A}, \quad \text{for all } \alpha, \beta \in \mathbb{R}.$$

The set $\{a_1, \dots, a_n\}$ is a *basis* for the subspace $\mathscr{A}$ if the following hold:

- $a_1, \dots, a_n$ *span* $\mathscr{A}$, *i.e.*,

$$\mathscr{A} = \operatorname{span}(a_1, \dots, a_n) := \{x_1 a_1 + \cdots + x_n a_n \mid x_1, \dots, x_n \in \mathbb{R}\} = \{\begin{bmatrix} a_1 & \cdots & a_n \end{bmatrix} x \mid x \in \mathbb{R}^n\}$$

- $\{a_1, \dots, a_n\}$ is an independent set of vectors.

Exercise problem 20. The number of basis vectors does not depend on the choice of the basis

□

The number of basis vectors of a subspace is invariant of the choice of the basis and is called the *dimension* of the subspace. The dimension of $\mathscr{A}$ is denoted by $\dim(\mathscr{A})$.

The *kernel* (also called *null space*) of the matrix $A \in \mathbb{R}^{m \times n}$ is the set of vectors mapped to zero by $f(x) := Ax$, *i.e.*,

$$\ker(A) := \{x \in \mathbb{R}^n \mid Ax = 0\}.$$

Adding a vector z in the kernel of A to a solution x_{p} of the system $Ax = y$ produces another solution of the system, *i.e.*, if $Ax_{\mathrm{p}} = y$, then $y = A(x_{\mathrm{p}} + z)$, for all $z \in \ker(A)$. From a parameter estimation point of view, $\ker(A)$ is the uncertainty in finding the parameter x, given the observation y. From a control point of view, $\ker(A)$ is the freedom in the control x that achieves the desired output y. If $\ker(A) = \{0\}$, the function $f(x) := Ax$ is called *one-to-one* .

Exercise problem 21. Show that $\ker(A) = \{0\}$ if and only if A is full column rank.

□

The image (also called column span or range) of the matrix $A^{m \times n}$ is the set of vectors that can be obtained as an output of the function $f(x) := Ax$, *i.e.*,

$$\operatorname{image}(A) := \{Ax \in \mathbb{R}^m \mid x \in \mathbb{R}^n\}.$$

Obviously, $\operatorname{image}(A)$ is the span of the columns of A. Alternatively, $\operatorname{image}(A)$ is the set of vectors y for which the system $Ax = y$ has a solution. If $\operatorname{image}(A) = \mathbb{R}^m$, the function $f(x) := Ax$ is called *onto*.

Exercise problem 22. Show that $\operatorname{image}(A) = \mathbb{R}^m$ if and only if A is full row rank.

□

For a matrix $A \in \mathbb{R}^{m \times n}$, $\ker(A)$ is a subspace of $\mathbb{R}^n$ and $\operatorname{image}(A)$ is a subspace of $\mathbb{R}^n$.

Exercise problem 23. Show that

$$\dim\big(\operatorname{image}(A)\big) = \operatorname{rank}(A) \qquad \text{and} \qquad \operatorname{col\,dim}(A) - \dim\big(\ker(A)\big) = \operatorname{rank}(A). \tag{1.6}$$

□

A direct consequence of (1.6) is the so called preservation of dimensions theorem (in $\mathbb{R}^n$)

$$\dim\big(\ker(A)\big) + \dim\big(\operatorname{image}(A)\big) = \operatorname{col\,dim}(A).$$

Note that

$$\operatorname{rank}(A) = \dim\big(\operatorname{image}(A)\big) = \operatorname{rank}(A^\top) = \dim\big(\operatorname{image}(A^\top)\big).$$

$\operatorname{image}(A)$ is the span of the columns of A and $\operatorname{image}(A^\top)$ is the span of the rows of A. The former is a subspace of $\mathbb{R}^m$ and the latter is a subspace of $\mathbb{R}^n$, they are equal (to the rank of A). By defining the *left kernel* of A,

$$\operatorname{left\,ker}(A) := \{y \in \mathbb{R}^m \mid y^\top A = 0\},$$

we have a preservation of dimensions theorem for $\mathbb{R}^m$

$$\dim\big(\operatorname{left}\ker(A)\big) + \dim\big(\operatorname{image}(A^\top)\big) = \operatorname{row}\dim(A).$$

The *standard basis* vectors in $\mathbb{R}^n$ are the vectors

$$e_1 = \begin{bmatrix} 1\\0\\0\\ \vdots \\ 0\end{bmatrix}, \quad e_2 = \begin{bmatrix} 0\\1\\0\\ \vdots \\ 0\end{bmatrix}, \quad \ldots, \quad e_n = \begin{bmatrix} 0\\0\\ \vdots \\ 0 \\ 1\end{bmatrix}.$$

Note that $e_1,\ldots,e_n$ are the columns of the identity matrix I_n. The elements of a vector $x \in \mathbb{R}^n$ are the coordinates of x with respect to a basis understood from the context. The default basis is the standard basis $e_1,\ldots,e_n$. Suppose that a new bases is given by the columns $t_1,\ldots,t_n$ of a matrix $T \in \mathbb{R}^{n\times n}$. Since $\{t_1,\ldots,t_n\}$ is a basis, the set is linearly independent. Therefore, the matrix T is nonsingular. Vice verse, any nonsingular matrix $T \in \mathbb{R}^{n\times n}$ defines a basis for $\mathbb{R}^n$. Let the coordinates of x in the basis T be $\widetilde{x}_1,\ldots,\widetilde{x}_n$. Then

$$x = \widetilde{x}_1 t_1 + \cdots + \widetilde{x}_n t_n = T\widetilde{x} \quad \Longrightarrow \quad \widetilde{x} = T^{-1}x,$$

i.e., the inverse matrix T^{-1} transforms the standard basis coordinates x into the T-basis coordinates $\widetilde{x}$.

Consider, now a linear operator $f : \mathbb{R}^n \to \mathbb{R}^n$ (*i.e.*, a function mapping from a space to the same space), given by $f(x) = Ax$, $A \in \mathbb{R}^{n\times n}$. The matrix A is a representation of f in a basis that is understood from the context. By default this is the standard basis. Changing the standard basis to a basis defined by the columns of a nonsingular matrix $T \in \mathbb{R}^{n\times n}$, the matrix representation of f changes to $T^{-1}AT$, *i.e.*,

$$\widetilde{y} = (T^{-1}AT)\widetilde{x}. \tag{1.7}$$

The mapping $A \mapsto T^{-1}AT$, defined by T, is called a *similarity transformation* of A.

1.5 Eigenvalues and eigenvectors

The (complex) number $\lambda \in \mathbb{C}$ is an *eigenvalue* of the square matrix $A \in \mathbb{R}^{n\times n}$ if there is a (complex) nonzero vector $v \in \mathbb{C}^n$, called an *eigenvector* associated to λ, such that $Av = \lambda v$. Equivalently, λ is an eigenvalue of A if the matrix $\lambda I_n - A$ is singular. If (λ, v) is an eigenpair of A, the action of A on vectors in $\operatorname{span}(v)$ is equivalent to a scalar multiplication by λ.

The *characteristic polynomial* of A is

$$p_A(\lambda) := \det(\lambda I_n - A).$$

The degree of p_A, denoted $\deg(p_A)$, is equal to n and p is *monic*, *i.e.*, the coefficient of the highest order term is one.

Exercise problem 24. Prove that the scalar $\lambda \in \mathbb{C}$ is an eigenvalue of A if and only if λ is a root of p_A.

□

The *geometric multiplicity* of λ is the dimension of the kernel of $\lambda I_n - A$. The *algebraic multiplicity* of λ is the multiplicity of the root λ of p_A. A matrix that has an eigenvalue for which the geometric and algebraic multiplicities do not coincide is called *defective*.

Suppose that $\{v_1,\ldots,v_n\}$ is a linearly independent set of eigenvectors of $A \in \mathbb{R}^{n\times n}$, *i.e.*,

$$Av_i = \lambda_i v_i, \qquad \text{for } i = 1,\ldots,n.$$

Written in a matrix form, the above set of equations is

$$A\underbrace{\begin{bmatrix} v_1 & \cdots & v_n\end{bmatrix}}_{V} = \underbrace{\begin{bmatrix} v_1 & \cdots & v_n\end{bmatrix}}_{V} \underbrace{\begin{bmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_n\end{bmatrix}}_{\Lambda}.$$

The matrix V has as columns the eigenvectors and is nonsingular since by assumption $\{v_1,\dots,v_n\}$ is a linearly independent set. Then

$$AV = V\Lambda \quad \implies \quad V^{-1}AV = \Lambda,$$

i.e., we obtain a similarity transformation (defined by the matrix $T := V^{-1}$) that *diagonalizes* A. Conversely, if there is a nonsingular $V \in \mathbb{C}^{n\times n}$, such that

$$V^{-1}AV = \Lambda$$

then $Av_i = \lambda_i v_i$, for $i = 1,\dots,n$, and therefore $\{v_1,\dots,v_n\}$ is a linearly independent set of eigenvectors.

The matrix $A \in \mathbb{R}^{n\times n}$ is *diagonalizable* if

- there is a nonsingular matrix T, such that TAT^{-1} is diagonal, or
- there is a set of n linearly independent eigenvectors of A.

The set of *defective* matrices corresponds to the set of matrices that are not diagonalizable. The eigenvalues of a matrix being distinct implies that the matrix is diagonalizable, however, the converse is not true (consider, for example, the the identity matrix). A prototypical example of a defective matrix is what is called the *Jordan block*

$$J_\lambda := \begin{bmatrix} \lambda & 1 & & \\ & \lambda & \ddots & \\ & & \ddots & 1 \\ & & & \lambda \end{bmatrix}.$$

A generalization of the eigenvalue decomposition $TAT^{-1} = \Lambda$ for defective matrices is the Jordan canonical form

$$TAT^{-1} = \operatorname{diag}(J_{\lambda_1},\dots,J_{\lambda_k}),$$

where $\lambda_1,,\dots,\lambda_k$ are the distinct eigenvalues of A.

Exercise problem 25. Show that the eigenvalues of a symmetric matrix $A \in \mathbb{R}^{n\times n}$ are real and the eigenvectors can be chosen to form an orthonormal set, *i.e.*, be orthogonal to each other and have unit norm.

□

1.6 Summary

- f is *linear* if homogeneity and superposition holds, *i.e.*, $f(\alpha x + \beta v) = \alpha f(x) + \beta f(v)$
- f is linear if and only if there is a matrix A, such that $f(x) = Ax$
- *image* (column span or range) of $A \in \mathbb{R}^{m\times n}$ is the set $\operatorname{image}(A) := \{Ax \mid x \in \mathbb{R}^n\}$
- *kernel* (or null space) of $A \in \mathbb{R}^{m\times n}$ is the set $\ker(A) := \{x \in \mathbb{R}^m \mid Ax = 0\}$
- $\mathscr{A} \subset \mathbb{R}^n$ is a *subspace* if $\alpha a + \beta b \in \mathscr{A}$ for all $a, b \in \mathscr{A}$
- a *basis* of a subspace is a set of linearly independent vectors that span the subspace
- the *dimension* of a subspace is the number of basis vectors
- the $\operatorname{image}(A)$ and the $\ker(A)$ of any matrix A are subspaces
- the *rank* of A is the number of linearly independent rows (or columns)
- $\dim\big(\operatorname{image}(A)\big) = \operatorname{rank}(A)$ and $\operatorname{coldim}(A) - \dim\big(\ker(A)\big) = \operatorname{rank}(A)$
- A is *full row rank* if $\operatorname{rank}(A) = \operatorname{rowdim}(A)$

- A is *full column rank* if $\operatorname{rank}(A) = \operatorname{col}\dim(A)$
- A is *full rank* if A is either full row rank or full column rank
- A is *nonsingular* if A is square and full rank
- *inversion problem:* given $y = Ax$, find x
- A^{-1} is *inverse* of A if $A^{-1}A = A^{-1}A = I$
- for A to have inverse, A should be square and full rank
- A^{L} is *left inverse* of A if $A^{\mathrm{L}}A = I$
- solution of the inversion problem: $x = A^{\mathrm{L}}y$, $A^{\mathrm{L}}A = I$
- left inverse exists if and only if A is full column rank
- *least-squares left inverse* $A^{\mathrm{L}}_{\mathrm{ls}} = (A^\top A)^{-1}A^\top$
- A^{R} is *right inverse* of A if $AA^{\mathrm{R}} = I$
- right inverse exists if and only if A is full row rank (nonsingular)
- *least-norm right inverse* $A^{\mathrm{R}}_{\mathrm{ln}} = A^\top(AA^\top)^{-1}$
- *2-norm* of a vector $\|x\| = \sqrt{x^\top x}$, *unit ball* $\{x \mid \|x\| \leq 1\}$
- *inner product* of $x, y \in \mathbb{R}^n$ is the scalar $\langle x, y\rangle := x^\top y$
- *Cauchy-Schwarz inequality:* $|a^\top b| \leq \|a\|\|b\|$
- $x, y \in \mathbb{R}^n$ are *orthogonal*, defined by $x \perp y$, if $\langle x, y\rangle = 0$
- *similarity transformation:* $A \mapsto T^{-1}AT$, T nonsingular
- *eigenvalue decomposition:* $T^{-1}AT = \Lambda$, Λ diagonal
- *characteristic polynomial of A:* $p_A(\lambda) := \det(\lambda I - A)$
- *symmetric matrix* $A = A^\top \implies \begin{array}{l}\text{real eigenvalues}\\ \text{orthonormal eigenvectors}\end{array}$

Notes and references

Excellent undergraduate level introduction to linear algebra is the text of G. Strang [Str98]. A more advanced text on linear algebra by the same author is [Str76]. The text of Meyer [Mey00] is encyclopedic and is well written. Advanced topics on matrix theory are covered by Horn and Johnson in [HJ85, HJ91].

Bibliography

[HJ85] R. Horn and C. Johnson. *Matrix Analysis.* Cambridge University Press, 1985.

[HJ91] R. Horn and C. Johnson. *Topics in Matrix Analysis.* Cambridge University Press, 1991.

[Mey00] C. Meyer. *Matrix Analysis and Applied Linear Algebra.* SIAM, 2000.

[Str76] G. Strang. *Linear Algebra and Its Applications.* Academic Press, 1976.

[Str98] G. Strang. *Introduction to Linear Algebra.* Wellesley College, 1998.

Chapter 2

Numerical linear algebra

- Projectors, Gram-Schmidt, and QR factorization
- Computation of eigenvalues and eigenvectors
- Singular value decomposition
- Conditioning of a problem
- Floating point arithmetic and stability of an algorithm

2.1 Projectors, Gram-Schmidt, and QR factorization

Projectors

Consider a finite set of vectors $\mathscr{Q} := \{q_1, \dots, q_k\} \subset \mathbb{R}^n$.

- $\mathscr{Q}$ is *normalized* if $\|q_i\| = 1$, for $i = 1, \dots, k$.
- $\mathscr{Q}$ is *orthogonal* if $q_i \perp q_j$, for all $i \neq j$.
- $\mathscr{Q}$ is *orthonormal* if $\mathscr{Q}$ is orthogonal and normalized.

Exercise problem 26. Show that an orthonormal set of vectors is independent.

□

Define the matrix $Q := \begin{bmatrix} q_1 & \cdots & q_k \end{bmatrix}$ and note that $\mathscr{Q}$ is orthonormal if and only if $Q^\top Q = I_k$. A matrix Q, such that $Q^\top Q = I_k$, is sometimes called *orthonormal* (this is not a completely standard terminology).

Exercise problem 27. Show that

- multiplication with an orthonormal matrix Q *preserves norm*, *i.e.*, $\|Qz\|^2 = z^\top Q^\top Qz = \|z\|^2$, and
- multiplication with an orthonormal matrix Q *preserves inner product*, *i.e.*, $\langle Qz, Qy\rangle = \langle z, y\rangle$.

□

Consider an orthonormal matrix $Q \in \mathbb{R}^{n\times k}$ and the subspace $\mathscr{L}$ spanned by the columns of Q, *i.e.*,

$$\mathscr{L} := \operatorname{image}(Q) \subseteq \mathbb{R}^n.$$

The columns of Q form an *orthonormal basis* for $\mathscr{L}$. Since Q is orthonormal, $Q^\top Q = I_k$, however, for $k < n$,

$$QQ^\top \neq I_n.$$

Exercise problem 28. The matrix $\Pi_{\operatorname{image}(Q)} := QQ^\top$ is an *orthogonal projector* onto $\mathscr{L}$, *i.e.*,

$$\Pi_{\mathscr{L}} x = \arg\min_y \|x - y\|_2 \quad \text{subject to} \quad y \in \mathscr{L} \tag{2.1}$$

$\square$

Exercise problem 29. Show that necessary and sufficient conditions for Π to be a projector are

1. $\Pi = \Pi^2$ (Π is idempotent), and
2. $\Pi = \Pi^\top$ (Π is symmetric).

$\square$

Exercise problem 30. Show that the matrix

$$\Pi^\perp := I - \Pi$$

is also an orthogonal projector. The projector $\Pi^\perp$ is called the *complementary projector* to Π. Define for a set $\mathscr{S} \subset \mathbb{R}^n$, its *orthogonal complement*

$$\mathscr{S}^\perp := \{x \in \mathbb{R}^n \mid x^\top y = 0 \text{ for all } y \in \mathscr{S}\}.$$

Show that for any set $\mathscr{S} \subset \mathbb{R}^n$, its orthogonal complement $\mathscr{S}^\perp$ is a subspace and that $\Pi^\perp$ projects onto the orthogonal complement $\big(\operatorname{image}(\Pi)\big)^\perp$ of $\operatorname{image}(\Pi)$.

$\square$

If $\mathscr{Q} := \{q_1, \dots, q_k\} \subset \mathbb{R}^n$ is an *orthonormal set* of $k = n$ vectors, then the matrix $Q := \begin{bmatrix} q_1 & \cdots & q_n \end{bmatrix}$ is called *orthogonal*. A matrix $Q \in \mathbb{R}^n$ is orthogonal if and only if it satisfies the identities

$$Q^\top Q = QQ^\top = \sum_{i=1}^n q_i q_i^\top = I_n.$$

It follows that for an orthogonal matrix Q, $Q^{-1} = Q^\top$. The identity $x = QQ^\top x$ is an expansion of x in the orthonormal basis given by the columns of Q.

- $\widetilde{x} := Q^\top x$ is the vector of the coordinates of x in the basis $\mathscr{Q}$, and
- $x = Q\widetilde{x}$ reconstructs x in the standard basis $\{e_1, \dots, e_n\}$.

Geometrically multiplication by Q (and $Q^\top$) is a rotation.

Exercise problem 31. Verify that a matrix representation of rotation in $\mathbb{R}^2$ is an orthogonal matrix.

$\square$

Gram-Schmidt procedure and QR factorization

Given an independent set of vectors $\{a_1, \dots, a_k\} \subset \mathbb{R}^n$, the Gram-Schmidt procedure, see Algorithm 1, produces an orthonormal set $\{q_1, \dots, q_k\} \subset \mathbb{R}^n$, such that

$$\operatorname{span}(a_1, \dots, a_r) = \operatorname{span}(q_1, \dots, q_r), \qquad \text{for all } r \leq k.$$

Algorithm 1 Gram-Schmidt

```
Input: $\{a_1, \dots, a_k\} \subset \mathbb{R}^n$
1: $q_1 := a_1/\|a_1\|$
2: for $i = 2, \dots, k$ do
3:     $v_i := (I - \Pi_{\operatorname{image}(q_1,\dots,q_{i-1})})a_i$    {project $a_i$ onto $\big(\operatorname{span}(q_1,\dots,q_{i-1})\big)^\perp$}
4:     $q_i := v_i/\|v_i\|$    {normalize}
5: end for
Output: $\{q_1, \dots, q_k\} \subset \mathbb{R}^n$
```

From $a_i \in \operatorname{span}(q_1, \dots, q_k)$, it follows that

$$a_i = r_{1i}q_1 + \cdots + r_{ii}q_i \tag{2.2}$$

for some scalars r_{ij}, for $i \leq j$ and $j = 1, \dots, k$.

Exercise problem 32. Show that $r_{ij} = q_i^\top a_j$.

□

Written in a matrix form (2.2) is what is called the QR matrix factorization

$$\underbrace{\begin{bmatrix} a_1 & a_2 & \cdots & a_k \end{bmatrix}}_{A} = \underbrace{\begin{bmatrix} q_1 & q_1 & \cdots & q_k \end{bmatrix}}_{Q_1} \underbrace{\begin{bmatrix} r_{11} & r_{12} & \cdots & r_{1k} \\ 0 & r_{22} & \cdots & r_{2k} \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & r_{kk} \end{bmatrix}}_{R_1}.$$

Here $Q_1 \in \mathbb{R}^{n\times k}$ is orthonormal and $R_1 \in \mathbb{R}^{k\times k}$ is upper triangular.

The Gram-Schmidt procedure applies a sequence of linear operations on the columns of A, such that the resulting matrix has orthonormal columns. In a matrix form this is expressed by $A\widetilde{R}_1 = Q_1$, where the matrix $\widetilde{R}_1$ encodes the operations of the Gram-Schmidt procedure. Note that $\widetilde{R}_1 = R_1^{-1}$, where R_1 is the upper triangular matrix of the QR factorization $A = Q_1R_1$.

Exercise problem 33. Show that the inverse $\widetilde{R}_1$ of a nonsingular upper triangular matrix R_1 is again upper triangular

□

Therefore, the Gram-Schmidt procedure can be termed "triangular orthogonalization".

There is an alternative procedure for computing the QR factorization that applies a sequence of orthogonal transformations on the columns of A, aiming to produce an upper triangular matrix. In a matrix form this is expressed by $\widetilde{Q}_1 A = R_1$, where Q_1

$$\widetilde{Q}_1 = \widetilde{Q}^{(1)} \cdots \widetilde{Q}^{(n)}$$

encodes the sequence of orthogonal transformations.

Exercise problem 34. Show that the product of orthonormal matrices $\widetilde{Q}^{(1)}, \dots, \widetilde{Q}^{(n)}$ is orthonormal.

□

$\widetilde{Q} = Q_1^\top$ is the orthogonal matrix of the QR factorization $A = Q_1R_1$. The alternative process of "orthogonal triangularization" turns our to be numerically more stable than the Gram-Schmidt procedure and is the preferred way of computing the QR factorization

If $\{a_1, \dots, a_k\}$ is a set of dependent vectors, $v_i := (I - \Pi_{\text{span}(q_1,\dots,q_{i-1})})a_i = 0$ for some i. Conversely, if $v_i = 0$ for some i, a_i is linearly dependent on $a_1, \dots, a_{i-1}$. The Gram-Schmidt produces can handle linearly dependent sets by skipping to the next input vector a_{i+1} whenever $v_i = 0$. As a result, the matrix R_1 is in *upper staircase form*, *e.g.*,

$$\begin{bmatrix} \times & \times & \times & \times & \times & \times & \times \\ & \times & \times & \times & \times & \times & \times \\ & & & \times & \times & \times & \times \\ & & & & & & \times \\ & & & & & & \\ & & & & & & \\ & & & & & & \end{bmatrix}, \tag{2.3}$$

with all empty elements being zeros.

Exercise problem 35. Which vectors are linearly dependent in the example of (2.3)?

□

The factorization

$$A = \begin{bmatrix} Q_1 & Q_2 \end{bmatrix} \begin{bmatrix} R_1 \\ 0 \end{bmatrix} = QR$$

with $Q := \begin{bmatrix} Q_1 & Q_2 \end{bmatrix}$ orthogonal and R_1 upper triangular is called *full QR factorization*.

Exercise problem 36. Show that

- $\text{image}(A) = \text{image}(Q_1)$, and

- $\big(\text{image}(A)\big)^{\perp} = \text{image}(Q_2)$.

$\square$

A method for finding the full QR factorization of A is to complete A to a full rank matrix, *e.g.*, $A_{\text{m}} := \begin{bmatrix} A & I \end{bmatrix}$, and apply the Gram-Schmidt procedure on A_{m}. In Matlab, `[Q,R] = qr(A)` procedures the full QR and `[Q1,R1] = qr(A,0)` procedures the reduced QR.

2.2 Computation of eigenvalues and eigenvectors

Three applications of the eigenvalue decomposition

- *Evaluation of a function $f : \mathbb{R} \to \mathbb{R}$ at a square matrix.* Consider a real analytic function $x \mapsto f(x)$ with a Taylor series expansion
$$f(x) = \frac{1}{0!} f(0)x^0 + \frac{1}{1!}\big(\tfrac{\mathrm{d}}{\mathrm{d}t} f\big)(0)x^1 + \frac{1}{2!}\big(\tfrac{\mathrm{d}^2}{\mathrm{d}t^2} f\big)(0)x^2 + \cdots$$
A matrix valued matrix function $X \mapsto f(X)$, where $X \in \mathbb{R}^{n\times n}$, corresponding to the scalar valued scalar function $x \mapsto f(x)$, where $x \in \mathbb{R}$ is defined by the series
$$f(X) := \frac{1}{0!} f(0)X^0 + \frac{1}{1!}\big(\tfrac{\mathrm{d}}{\mathrm{d}t} f\big)(0)X^1 + \frac{1}{2!}\big(\tfrac{\mathrm{d}^2}{\mathrm{d}t^2} f\big)(0)X^2 + \cdots$$
Exercise problem 37. Assuming that X is diagonalizable, *i.e.*, there is a nonsingular matrix V and a diagonal matrix Λ, such that
$$X = V^{-1}\Lambda V, \qquad \Lambda = \text{diag}(\lambda_1, \dots, \lambda_n),$$
show that
$$f(X) = V \begin{bmatrix} f(\lambda_1) & & \\ & \ddots & \\ & & f(\lambda_n) \end{bmatrix} V^{-1} =: V f(\Lambda) V^{-1}.$$
$\square$

- *Stability of linear time-invariant systems.* Consider the first order vector linear constant coefficients differential and difference equations
$$\tfrac{\mathrm{d}}{\mathrm{d}t} x(t) = Ax(t), \text{ for } t \in \mathbb{R}_+ \qquad \text{and} \qquad x(t+1) = Ax(t), \text{ for } t \in \mathbb{Z}_+. \tag{2.4}$$
Given $x(0) \in \mathbb{R}^n$, the equations (2.4) have unique solutions x. Qualitative properties of the set of solutions, such as, *stability*, *i.e.*,
$$x(t) \to 0 \quad \text{as} \quad t \to \infty,$$
are determined by the *location of the eigenvalues of A*.
Exercise problem 38. Show that for the differential equation (continuous-time system), stability holds if and only if $\Re(\lambda_i) < 0$ for all i and for the difference equation (discrete-time system), stability holds if $|\lambda_i| < 1$ for all i.
$\square$
Stability of (2.4), however, can be checked without computing the eigenvalues $\lambda_1, \dots, \lambda_n$, *cf.*, the Routh–Hurwitz (continuous-time systems) and Schur–Cohn (discrete-time systems) tests [Jur74].

- *Principal component analysis (PCA).* Principal component analysis is a statistical technique. Here we give an alternative deterministic formulation. Given a set of vectors $\{a_1, \dots, a_n\}$, find
$$\{b_1^*, \dots, b_j^*\} := \arg\max_{b_1,\dots,b_j} \left\| \Pi_{\text{span}(b_1,\dots,b_j)} \begin{bmatrix} a_1 & \cdots & a_n \end{bmatrix} \right\|_{\text{F}} \qquad \text{subject to} \qquad \begin{bmatrix} b_1 & \cdots & b_j^* \end{bmatrix}^{\top} \begin{bmatrix} b_1 & \cdots & b_j^* \end{bmatrix} = I \tag{2.5}$$
(Recall that $\Pi_{\text{span}(b_1,\dots,b_j)}$ is the orthogonal projector onto $\text{span}(b_1, \dots, b_j)$.)

Exercise problem 39. Shown that the solution $\{b_1^*,\dots,b_n^*\}$ to (2.5) is the orthonormal set of eigenvectors $\{v_1,\dots,v_j\}$ of the matrix $A^\top A$, corresponding to the largest eigenvalues.

$\square$

Eigenvalue/eigenvector computation algorithms

Eigenvalue computation is closely related to rooting a polynomial.

Exercise problem 40. Show that the set of eigenvalues of the *companion matrix*

$$C_p := \begin{bmatrix} -p_{n-1} & -p_{n-2} & \cdots & -p_1 & -p_0 \\ 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & \ddots & \vdots & \vdots \\ \vdots & \ddots & \ddots & 0 & 0 \\ 0 & \cdots & 0 & 1 & 0 \end{bmatrix}$$

coincides with the set of roots of a polynomial

$$p(z) = p_0 + p_1 z + \cdots + p_n z^n.$$

$\square$

A classical results in mathematics due to Abel, is that there is no analogue of the formula for the roots of a quadratic polynomial for a general polynomial of degree more than 4. By the above link between roots of a polynomial and eigenvalues of a matrix, we conclude that the same must be true for the eigenvalues of a general matrix of dimension more than 4×4.

There is no algorithm that can compute the eigenvalues of a general matrix of dimension more than 4×4 in a finite number of operations.

Eigenvalue algorithms must be *iterative* and in finite time produce only an *approximation* of the eigenvalues. The aim in the design of eigenvalue algorithms is to increase the converges speed and reduce the number of operations per iteration, so that the sequence of the eigenvalue approximations produced by the algorithm converges rapidly to the eigenvalues.

The power iteration, inverse power iteration, and Rayleigh quotient iteration are basic methods for computing an eigenvalue/eigenvector pair of a matrix and are ingredients of the modern algorithms for eigenvalue computation. Next we outline these algorithms and for simplicity we restrict to the symmetric case. A symmetric $A \in \mathbb{R}^{n\times n}$ has n real eigenvalues, which we index as follows

$$\lambda_{\max} := \lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n =: \lambda_{\min}.$$

Corresponding to $\lambda_1,\dots,\lambda_n$, we choose an orthonormal set of eigenvectors $q_1,\dots,q_n$. The Rayleigh quotient of $v \in \mathbb{R}^n$ (with respect to A) is a mapping $r : \mathbb{R}^n \to \mathbb{R}$ defined by

$$r(v) := \frac{v^\top A v}{v^\top v}.$$

Note that $r(\alpha q_i) = \lambda_i$, for all $\alpha \in \mathbb{R}$ and $i = 1,\dots,n$.

Exercise problem 41. Show that $\min_v r(v) = \lambda_{\min}$ and $\max_v r(v) = \lambda_{\max}$.

$\square$

Exercise problem 42. Show that if $|\lambda_1| > |\lambda_2|$ and $v_1^\top v^{(0)} \neq 0$, then $v^{(k)} \to \pm v_1$ with *linear convergence* rate $O(|\lambda_2/\lambda_1|)$.

$\square$

Let λ be the closest eigenvalue to μ and λ' be the second closest. Let v be the unit norm eigenvector corresponding to λ. Then if $v^\top v^{(0)} \neq 0$, then $v^{(k)} \to \pm v$ with *linear convergence* rate $O(|(\mu-\lambda')/(\mu-\lambda)|)$.

Let λ be the closest eigenvalue to μ and v be the corresponding eigenvector. Then, if $v^\top v^{(0)} \neq 0$, then $v^{(k)} \to \pm v$ with *cubic convergence rate.*

Algorithm 2 Power iteration.

```
Input: unit norm vector $v^{(0)}$ and a symmetric matrix $A$.
1: for $k = 1, 2, \dots$ (till convergence) do
2:     $w := Av^{(k-1)}$    {apply $A$}
3:     $v^{(k)} := w/\|w\|$    {normalize}
4: end for
Output: eigenvalue/eigenvector of $A$ — $\big((v^{(k)})^\top Av^{(k)}, v^{(k)}\big)$
```

Algorithm 3 Inverse iteration.

```
Input: unit norm vector $v^{(0)}$, a symmetric matrix $A$, and $\mu \geq 0$.
1: for $k = 1, 2, \dots$ (till convergence) do
2:     $(A - \mu I)w = v^{(k-1)}$    {apply $(A - \mu I)^{-1}$}
3:     $v^{(k)} := w/\|w\|$    {normalize}
4: end for
Output: eigenvalue/eigenvector of $A$ — $\big((v^{(k)})^\top Av^{(k)}, v^{(k)}\big)$
```

Exercise problem 43. Implement the power, inverse power, and Rayleigh quotient methods. Apply them on examples and observe their convergence properties. Comment on the results.

□

Normalized simultaneous power iteration

Take a set of initial vectors $\{v_1^{(0)}, \dots, v_p^{(0)}\}$ and consider the iteration:

$$\underbrace{\begin{bmatrix} v_1^{(k+1)} & \cdots & v_p^{(k+1)} \end{bmatrix}}_{V^{(k+1)}} = A \underbrace{\begin{bmatrix} v_1^{(k)} & \cdots & v_p^{(k)} \end{bmatrix}}_{V^{(k)}}, \qquad k = 0, 1, \dots$$

One can expect that under suitable assumptions

$$\operatorname{span}(v_1^{(k)}, \dots, v_p^{(k)}) \to \operatorname{span}(v_1, \dots, v_p), \qquad \text{as} \quad k \to \infty.$$

However,

$$v_i^{(k)} \to v_1, \qquad \text{as} \quad k \to \infty, \quad \text{for all } i,$$

so $V^{(k+1)}$ becomes increasingly *ill-conditioned* as $k \to \infty$. This problem is resolved by changing the computed basis on each iteration step to an equivalent orthonormal basis.

Under suitable assumptions

$$\operatorname{image}(Q^{(k)}) \to \operatorname{span}(v_1, \dots, v_p), \qquad \text{as} \quad k \to \infty.$$

Algorithm 4 Rayleigh quotient iteration

```
Input: unit norm vector $v^{(0)}$ and symmetric matrix $A$
1: for $k = 1, 2, \dots$ (till convergence) do
2:     $(A - \lambda^{(k-1)} I)w = v^{(k-1)}$    {apply $(A - \lambda^{(k-1)} I)^{-1}$}
3:     $v^{(k)} := w/\|w\|$    {normalize}
4:     $\lambda^{(k)} := (v^{(k)})^\top Av^{(k)}$    {eigenvalue estimate}
5: end for
Output: eigenvalue/eigenvector of $A$ — $\big(\lambda^{(k)}, v^{(k)}\big)$
```

Algorithm 5 Normalized simultaneous power iteration.

```
Input: orthonormal matrix $Q^{(0)} \in \mathbb{R}^{n\times p}$ and symmetric matrix $A$.
1: for $k = 1,2,\ldots$ (till convergence) do
2:     $Z = AQ^{(k-1)}$        {apply $A$}
3:     QR factorization $Q^{(k)}R^{(k)} = Z$        {compute orthonormal basis for image$(Z)$}
4: end for
Output: orthonormal eigenvectors of $A$ — $Q^{(k)}$
```

QR algorithm

The basic QR algorithm is normalized simultaneous power iteration with a full set $p = n$ vectors and initial condition $Q^{(0)} = I_n$. Note that

$$A^{(k)} = R^{(k)}Q^{(k)} = Q^{(k)\top}A^{(k-1)}Q^{(k)},$$

so that $A^{(k)}$ is similar to $A^{(k-1)}$.

Additional features of a practical QR algorithm are

- *pre-processing:* reduce A to tridiagonal form before the iteration,
- *shifts:* factor $A^{(k)} - \lambda^{(k)}I$ instead of $A^{(k)}$, where $\lambda^{(k)}$ is an eigenvalue estimate, and
- *deflations:* reduce the size of A when and eigenvalue is found.

Algorithm 6 QR algorithm.

```
Input: symmetric matrix $A^{(0)} = A$.
1: for $k = 1,2,\ldots$ (till convergence) do
2:     $A^{(k-1)} = Q^{(k)}R^{(k)}$        {QR factorization}
3:     $A^{(k)} = R^{(k)}Q^{(k)}$        {recombine in reverse order}
4: end for
Output: a Schur decomposition of $A$ — $Q^{(k)}, R^{(k)}$.
```

The QR algorithm with shifts corresponds to the Rayleigh quotient iteration.

Note 44 (Hessenberg and Schur forms). The analogue of the tridiagonal form for nonsymmetric matrices is the Hessenberg form

$$H := \begin{bmatrix} \times & \times & \cdots & \times & \times \\ \times & \times & \cdots & \times & \times \\ & \times & \ddots & \ddots & \times \\ & & \ddots & \ddots & \vdots \\ & & & \times & \times \end{bmatrix}$$

There is an orthogonal similarity transformation that brings any square matrix to a Hessenberg form. Modern algorithms for eigenvalue factorization of a general matrix A consist of two distinct stages:

1. reduction of A by an orthogonal similarity transformation to *Hessenberg form*, and
2. iterative convergence by a sequence of unitary similarity transformation to a *Schur form*.

Let U^* be the conjugate transposed of $U \in \mathbb{C}^{m\times n}$. The matrix U is unitary if $U = U^*$. The Schur form R of A is an upper triangular matrix, such that $R = U^*AU$, where U is a unitary matrix. The Schur form is *eigenvalue revealing* because the eigenvalues of A appear on the diagonal of R. The first stage requires a finite number of operations and is introduced for the purpose of speeding up the convergence on the second stage, which requires an infinite number of operations.

Generalized eigenvalues

A pair (A,B) of square matrices $A,B \in \mathbb{R}^{n\times n}$ is called a *pencil*. A generalized eigenvector/eigenvalue (v,λ) of the pencil (A,B) is a vector $v \in \mathbb{C}^n$ and a scalar $\lambda \in \mathbb{C}$, such that

$$Av = \lambda Bv.$$

For a nonsingular matrix B, the generalized eigenvalue problem is equivalent to a standard eigenvalue problem

$$B^{-1}Av = \lambda v.$$

If A and B are symmetric matrices, the pencil (A,B) is called symmetric and has real generalized eigenvalues. The generalized Rayleigh quotient is a mapping $r : \mathbb{R}^n \to \mathbb{R}$ defined by

$$r_{A,B}(v) := \frac{v^\top Av}{v^\top Bv}$$

It has analogous properties to the Rayleigh quotient.

Exercise problem 45. Consider a symmetric pencil (A,B) and let $\lambda_{\min}/\lambda_{\max}$ be the minimal/maximal generalized eigenvalue of (A,B). Show that

$$\lambda_{\min} = \min_{v\in\mathbb{R}^n} r_{A,B}(v) \quad \text{and} \quad \lambda_{\max} = \max_{v\in\mathbb{R}^n} r_{A,B}(v).$$

□

2.3 Singular value decomposition

The singular value decomposition is used as both computational and analytical tool. Any matrix $A \in \mathbb{R}^{m\times n}$ has a singular value decomposition

$$A = \underbrace{\begin{bmatrix} u_1 & \cdots & u_r \end{bmatrix}}_{U_1} \underbrace{\begin{bmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_r \end{bmatrix}}_{\Sigma_1} \underbrace{\begin{bmatrix} v_1 & \cdots & v_r \end{bmatrix}^\top}_{V_1^\top}, \tag{2.6}$$

where U_1 and V_1 are orthonormal and $\sigma_1,\dots,\sigma_r$ are positive scalars, called *singular values* of A. The columns of U, $u_1,\dots,u_r$ are called *left singular vectors* and the columns of V, $v_1,\dots,v_r$ are called *right singular vectors* of A.

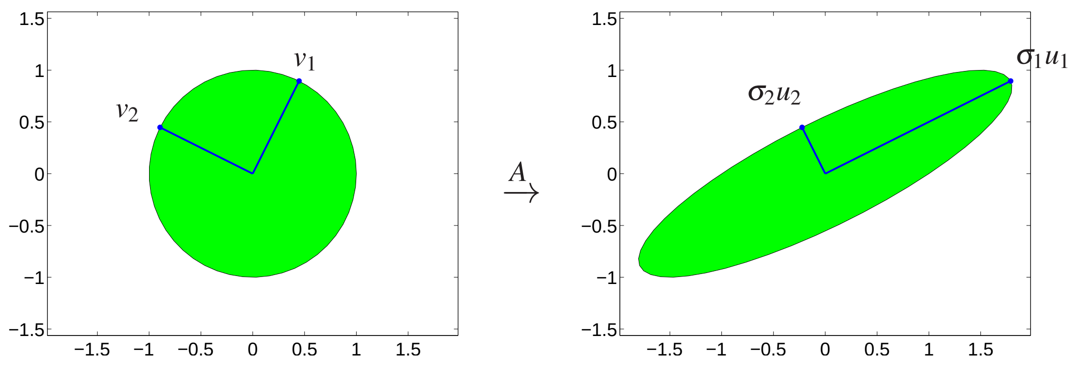
A geometric fact motivating the singular value decomposition is:

the image $\mathscr{E} = A\mathscr{U}$ of a unit ball $\mathscr{U}$ under a linear map $x \mapsto Ax$ is a hyperellips.

Example 46. Consider the following example

$$\underbrace{\begin{bmatrix} 1.00 & 1.50 \\ 0 & 1.00 \end{bmatrix}}_{A} = \underbrace{\begin{bmatrix} 0.89 & -0.45 \\ 0.45 & 0.89 \end{bmatrix}}_{U} \underbrace{\begin{bmatrix} 2.00 & 0 \\ 0 & 0.50 \end{bmatrix}}_{\Sigma} \underbrace{\begin{bmatrix} 0.45 & -0.89 \\ 0.89 & 0.45 \end{bmatrix}}_{V^\top}.$$

The image of the unit ball under the linear map defined by A is



The vector v_1 is mapped by A to the vector $\sigma_1 u_1$, and the vector v_2 is mapped by A to the vector $\sigma_2 u_2$. From all unit norm inputs vectors, the vector v_1 achieves the maximum 2-norm output vector $\sigma_1 u_1$, and v_2 achieves the minimum 2-norm output vector $\sigma_2 u_2$. Note that $\|\sigma_i u_i\| = \sigma_i$.

The decomposition (2.6) is sometimes called the *reduced SVD* of a matrix $A \in \mathbb{R}^{m\times n}$ in order to distinguish it from the *full SVD*

$$A = U\Sigma V,$$

where $U \in \mathbb{R}^{m\times m}$ and $V \in \mathbb{R}^{n\times n}$ are orthogonal and

$$\Sigma = \begin{bmatrix} \Sigma_1 & 0 \\ 0 & 0 \end{bmatrix} \in \mathbb{R}^{m\times n}.$$

Compared with the reduced SVD, the full SVD computes matrices $U_2 \in \mathbb{R}^{m\times(m-r)}$ and $V_2 \in \mathbb{R}^{n\times(n-r)}$, such that

$$U := \begin{bmatrix} U_1 & U_2 \end{bmatrix} \quad \text{and} \quad V := \begin{bmatrix} V_1 & V_2 \end{bmatrix}$$

are orthogonal and adds zero rows/columns to Σ_1 to form $\Sigma \in \mathbb{R}^{m\times n}$. Note that the singular values of A are $\sigma_1, \ldots, \sigma_r$ *and* $\min(m-r, n-r)$ zeros. In Matlab the full SVD is computed via `[U,S,V] = svd(A)` and the reduced SVD via `[U,S,V] = svd(A,0)`.

Similarities between SVD and EVD: Both the SVD and EVD diagonalize a matrix A. The left singular vectors of A are eigenvectors of $AA^\top$, the right singular vectors of A are eigenvectors of $A^\top A$, the nonzero singular values of A are the square roots of the nonzero eigenvalues of $AA^\top$ or $A^\top A$. In particular, for a symmetric A, $|\lambda_i| = \sigma_i$ and for a positive definite matrix $\lambda_i = \sigma_i$.

Exercise problem 47. Using the fact that $AA^\top$ or $A^\top A$ have the same nonzero eigenvalues, argue (without doing computations) that the eigenvalues of $\left[\begin{smallmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{smallmatrix}\right]$ are $3, 0, 0$.

$\square$

Differences between SVD and EVD: Despite the above similarities, there are important differences between the SVD and EVD. The SVD exists for *any matrix*, while the EVD exist for nondefective square matrices. The SVD applies *two orthogonal similarity transformations*, while the EVD applies one (in general not orthogonal) similarity transformation. The SVD is used to analyse a *single application of* A on a vector, while the EVD is useful in problems where A *is repeatedly applied*.

2.4 Conditioning and stability

Abstractly a computational problem is a mapping $f : \mathscr{X} \to \mathscr{Y}$ from a data space $\mathscr{X}$ to a solutions space $\mathscr{Y}$. Usually f is a continuous nonlinear function (even though the problem is in th realm of the linear algebra). A particular data instance is an element $X_0 \in \mathscr{X}$. The problem f is called *well conditioned* at the data X_0 if

$$\boxed{\text{small perturbations in } X_0 \text{ lead to small changes in } f(X_0)}$$

The *absolute condition number* of the problem f at the data instance X_0 is the derivative of f with respect to X at X_0

$$\lim_{\delta\to 0} \sup_{\|\widetilde{X}\|<\delta} \frac{\|f(X_0+\widetilde{X}) - f(X_0)\|}{\|\widetilde{X}\|}.$$

The *relative condition number* is defined as

$$\lim_{\delta\to 0} \sup_{\|\widetilde{X}\|<\delta} \frac{\|f(X_0+\widetilde{X}) - f(X_0)\|/\|f(X_0)\|}{\|\widetilde{X}\|/\|X_0\|}.$$

Example 48 (Conditioning of root finding). The roots finding problem is: Given polynomial coefficients $\{p_0, p_1, \dots, p_n\}$, find its roots $\{\lambda_1, \dots, \lambda_n\}$, *i.e.*,

$$p(\lambda) = p_0 + p_1\lambda^1 + \cdots + p_n\lambda^n = c(\lambda - \lambda_1)\cdots(\lambda - \lambda_n).$$

The relative condition number of λ_j with respect to perturbation $\widetilde{a}_i$ of a_i is

$$\kappa_{i,j} = |a_i\lambda_j^{i-1}| / |\tfrac{\mathrm{d}}{\mathrm{d}\lambda}p(\lambda_j)|.$$

For the polynomial $p(\lambda) = (\lambda - 1)\cdots(\lambda - 20)$, $\arg\max_{i,j}\kappa_{i,j} = (15, 15)$ and in Matlab we obtain

```
>> roots(poly([1:20]))

ans = 1.0000    ...    14.0684   14.9319   16.0509    ...    20.0003
```

The function `poly` gives the coefficients of a monic polynomial with roots, specified by the input argument. (This operation is recursive multiplication of a polynomial by monomials or equivalently convolution of a sequence by a sequence with length two.) The function `roots` solves the roots finding problem, *i.e.*, its argument is a vector of polynomial coefficients and the results is a vector of the roots. In exact arithmetic `roots(poly([1:20]))` should return the answer $\{1, \dots, 20\}$, however, due to the round-off errors in the finite precision arithmetic the answer is not exact. The experiment shows that the 15th root is the one that has the largest perturbation.

Exercise problem 49. Check the computed roots of $(\lambda - 1)^4$ (`roots(poly([1 1 1 1]))`).

Condition number of matrix–vector product

Theorem 50. *The problem of computing $y = Ax$ for given nonsingular matrix $A \in \mathbb{R}^{n\times n}$ and a vector $x \in \mathbb{R}^n$ has relative condition number with respect to perturbations in x*

$$\kappa = \|A\|\frac{\|x\|}{\|y\|} \leq \|A\|\|A^{-1}\|.$$

For a nonsingular matrix A, the number

$$\kappa(A) := \|A\|\|A^{-1}\|$$

is called the *condition number of A*. For a general (nonsquare) matrix and 2-norm $\|\cdot\|$,

$$\kappa(A) := \sigma_{\max}(A)/\sigma_{\min}(A).$$

The matrix A is called *ill-conditioned* if $\kappa(A)$ is “large”, and A is called *well-conditioned* if $\kappa(A)$ is “small”. (Here “large” and “small” depend on the size of the expected perturbations, so that they depend on the application.) The number $\kappa(A)$ is related to perturbations in the worst case. For an ill-conditioned A, the problem $y = Ax$ may still be well-conditioned at certain x’s.

Condition number of solving systems of equations

The relative condition number of the computational problem “solve a system of equations $y = Ax$, $A \in \mathbb{R}^{n\times n}$, with respect to perturbation in y” is given by Theorem 50. Indeed, provided A is nonsingular, $x = A^{-1}y$, which is a matrix–vector product problem. (The matrix now is A^{-1}.) If A is singular, the problem is infinitely ill-conditioned. Another term for this case is *ill-posed* problem.

Theorem 51. *The problem of computing $x = A^{-1}y$, given $A \in \mathbb{R}^{n\times n}$ and $y \in \mathbb{R}^n$ has relative condition number $\kappa(A)$ with respect to perturbations in A.*

Proof. The perturbation $\widetilde{A}$ in A leads to a perturbation $\widetilde{x}$ in x, such that

$$(A + \widetilde{A})(x + \widetilde{x}) = y \quad\Longrightarrow\quad \widetilde{A}x + A\widetilde{x} \overset{1}{=} 0.$$

Here "$\overset{1}{=}$" means equal up to first order terms in $\widetilde{A}$ and $\widetilde{x}$. ($\kappa(A)$ describes the effect of infinitesimal perturbations.)

$$\begin{aligned} \widetilde{x} \overset{1}{=} -A^{-1}\widetilde{A}x \quad &\implies \quad \|\widetilde{x}\| \le \|A^{-1}\|\|\widetilde{A}\|\|x\| \\ &\implies \quad \frac{\|\widetilde{x}\|/\|x\|}{\|\widetilde{A}\|/\|A\|} \le \|A^{-1}\|\|A\| = \kappa(A). \end{aligned}$$

$\square$

Digital representation of real numbers

The IEEE double precision arithmetic is a widely used standard for digital representation of real numbers. It is characterized by

- range: $[-2.23 \times 10^{-308}, 1.79 \times 10^{308}]$, and
- discretization: $[2^i, 2^{i+1}] \mapsto 2^i\{1, 1+2^{-52}, 1+2\times 2^{-52}, \ldots, 2\}$.

If a number large than 10^{308} is produced during computations, an exception called an *overflow* occurs. Conversely, if a number smaller than 10^{-308} is produced, an *underflow* occurs. In the case of underflow, the number is rounded to zero. Different systems react in different ways to overflow, which is a more serious exception.

The gaps between adjacent numbers are in a relative scale at most

$$\varepsilon := 2^{-52} \approx 2.22 \times 10^{-16}.$$

The number $\varepsilon := 2^{-52}$ is called the *machine precision*.

In *fixed point* arithmetic the position of the decimal point is fixed. In *floating point* arithmetic the position of the decimal point is stored together with the digits. Fixed point arithmetic leads to uniform absolute errors, while floating point arithmetic leads to uniform relative errors.

Let $\mathrm{fl}(x)$ be the digital representation of $x \in \mathbb{R}$, the error $|\mathrm{fl}(x) - x| \le \varepsilon$ is called *rounding error*.

Stability of an algorithm

Recall the general definition of a computational problem as a mapping $f : \mathscr{X} \to \mathscr{Y}$. A computational algorithm, aiming to solve the problem f, is another mapping $\widehat{f} : \mathscr{X} \to \mathscr{Y}$ from the data to the solution space. The algorithm $\widehat{f}$ is called *backward stable* if for each $X \in \mathscr{X}$ there is $\widehat{X} \in \mathscr{X}$, such that

$$\frac{\|X - \widehat{X}\|}{\|X\|} = O(\varepsilon) \quad \text{and} \quad \widehat{f}(X) = f(\widehat{X}),$$

i.e.,

$$\boxed{\text{backward stable algorithm gives the right answer to a nearby problem}}$$

The meaning of $e(\widetilde{X}) := \|\widetilde{X}\|/\|X\| = O(\varepsilon)$ is that there is $c, \delta > 0$, such that

$$\|\widetilde{X}\| < \delta \quad \implies \quad |e(\widetilde{X})| \le c\varepsilon.$$

Computational complexity of an algorithm

The computational complexity of an algorithm is measured by the number of floating point operations (flops) or by the execution time. One flop is defined as one addition, subtraction, multiplication, or division. The flops count is usually simplified to leading order terms, *e.g.*, $O(n)$. It is useful in theoretical comparison of algorithms but on modern computers it is not an accurate predictor of the computation time.

Standard linear algebra operations have the following computational complexities:

- n vector–vector operations (vector sum, scalar–vector multiplication, inner product) — $O(n)$ flops

- $m \times n$ matrix–vector product — $O(mn)$ flops
- $m \times n$ matrix – $n \times p$ matrix product — $O(mnp)$ flops
- solving $Ax = b$ with general $A \in \mathbb{R}^{n \times n}$ — $O(n^3)$ flops

However if the matrix A has special structure and this structure is exploited, the computational complexities may be lower. The following are typical examples of matrix structures and the corresponding computational complexities for solving systems of equations with such matrices:

- diagonal — n flops ($x_i = y_i/a_{ii}$ for $i = 1, \dots, n$)
- lower/upper triangular: n^2 flops (via forward/backward substitution)
- banded — $O(nk)$, where k is the bandwidth
- symmetric — $O(n^3/3)$ (via Cholesky decomposition)
- orthogonal — $O(n^2)$ ($x = A^T y$)
- permutation — 0 flops
- Toeplitz — $O(n^2)$ flops
- combination of banded, symmetric, and Toeplitz

Numerical linear algebra software

Currently the state-of-the-art software for numerical linear algebra are the Basic Linear Algebra Subroutines (BLAS) and Linear Algebra PACKage (LAPACK) libraries. BLAS has functions for the lower-level linear algebra operations, such as vector–vector operations (Level 1 BLAS), matrix–vector products (Level 2 BLAS), and matrix–matrix products (Level 3 BLAS). Although these operations are conceptually simple and algorithmically straightforward, their implementation on a computer with distributed memory, is a highly nontrivial task.

The LAPACK library contains functions for higher level operations such as matrix factorizations and solvers for linear systems, least-squares, and least-norm problems. There are special functions for structured problems involving triangular, banded, diagonal matrices and the solvers provide error bounds based on perturbation analysis.

Notes and references

A excellent concise introduction to numerical linear algebra is given by Trefethen and Bau [TB97]. Classic reference on this subject is the book of Golub and Van Loan [GV96]. The book of Demmel [Dem97] is with a particular emphasis on writing numerical software. Perturbation theory is treated in [SS90] and stability analysis is extensively treated by Higham [Hig91]. The IEEE floating point arithmetic is presented in [Ove01].

Bibliography

[Dem97] J Demmel. *Applied Numerical Linear Algebra.* SIAM, 1997.

[GV96] G. Golub and C. Van Loan. *Matrix Computations.* Johns Hopkins University Press, third edition, 1996.

[Hig91] N. Higham. *Accuracy and Stability of Numerical Methods.* SIAM, 1991.

[Jur74] E. Jury. *Inners and stability of dynamic systems.* Wiley, 1974.

[Ove01] M. Overton. *Numerical Computing with IEEE Floating Point Arithmetic.* SIAM, 2001.

[SS90] G. Stewart and J. Sun. *Matrix perturbation theory.* Academic Press, Boston, 1990.

[TB97] L. Trefethen and D. Bau. *Numerical Linear Algebra.* SIAM, 1997.

Chapter 3

Applications

- Least-squares
- Least-norm
- Total least-squares
- Low-rank approximation

The first three sections are discuss the linear system of equations $Ax = y$. The matrix $A \in \mathbb{R}^{m \times n}$ and the vector $y \in \mathbb{R}^m$ are given data. The vector $x \in \mathbb{R}^n$ is an unknown. Assuming that A is full rank, the system $Ax = y$ is called

- *overdetermined* if $m > n$ (in this case it has more equations than unknowns) and
- *underdetermined* if $m < n$ (in this case it has more unknowns than equations).

For most vectors $y \in \mathbb{R}^m$, an overdetermined system has no solution x, and for any $y \in \mathbb{R}^m$ an underdetermined system has infinitely many solutions x. In the case of an overdetermined system, it is of interested to find an approximate solution. An important example is the least squares approximate solution, which minimizes the 2-norm of the equation error.

In the case of an underdetermined system, it is of interested to find a particular solution. The least-norm solution is an example of a particular solution, It minimizes the 2-norm of the solution. Note that the least-squares approximate solution is (most of the time) not a solution, while the least-norm solution is (aways) one of infinitely many solutions.

3.1 Least-squares

The least-squares method for solving approximately an overdetermined system $Ax = y$ of equations is defined as follows. Choose x such that the 2-norm of the residual (equation error)

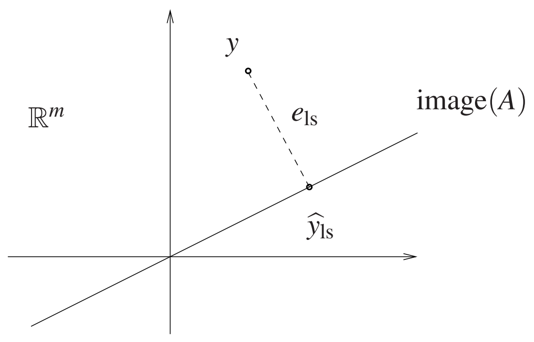
$$e(x) := y - Ax$$

is minimized. A minimizer

$$\widehat{x}_{\text{ls}} := \arg\min_x \| \underbrace{y - Ax}_{e(x)} \|_2 \tag{3.1}$$

is called a *least-squares approximate solution* of the system $Ax = y$.

A geometric interpretation of the least-squares approximation problem (3.1) projection of y onto the image of A.



Here $\widehat{y}_{\text{ls}} := A\widehat{x}_{\text{ls}}$ is the projection, which is the least-squares approximation of y and $e_{\text{ls}} := \widehat{y}_{\text{ls}} - A\widehat{x}_{\text{ls}}$ is the approximation error.

Let a_i be the ith row of A. We refer to the vector $\text{col}(a_i, y_i)$ as a "data point". We have,

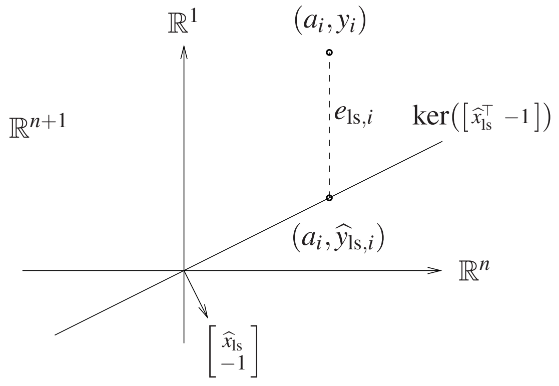
$$\begin{aligned} A\widehat{x}_{\text{ls}} = \widehat{y}_{\text{ls}} \quad &\iff \quad \begin{bmatrix} A & \widehat{y}_{\text{ls}} \end{bmatrix} \begin{bmatrix} \widehat{x}_{\text{ls}} \\ -1 \end{bmatrix} = 0 \\ &\iff \quad \begin{bmatrix} a_i & \widehat{y}_{\text{ls},i} \end{bmatrix} \begin{bmatrix} \widehat{x}_{\text{ls}} \\ -1 \end{bmatrix} = 0, \quad \text{for } i = 1, \dots, m \end{aligned}$$

so that for all i, $(a_i, \widehat{y}_{\text{ls},i})$ lies on the subspace perpendicular to $(\widehat{x}_{\text{ls}}, -1)$. $(a_i, \widehat{y}_{\text{ls},i})$ is an the least-squares approximation of the i data point $\text{col}(a_i, y_i)$.

$$(a_i, \widehat{y}_{\text{ls},i}) = (a_i, \widehat{y}_{\text{ls},i}) + (0, e_{\text{ls},i}),$$

and $(0, e_{\text{ls},i})$ is the least-squares approximation error. Note that $e_{\text{ls},i}$ is the vertical distance from (a_i, y_i) to the subspace.

The above derivation suggestions another geometric interpretation of the least-squares approximation.


$\mathbb{R}^1$
(a_i, y_i)
$\mathbb{R}^{n+1}$
$e_{\text{ls},i}$
$\ker([\widehat{x}_{\text{ls}}^\top \ -1])$
$(a_i, \widehat{y}_{\text{ls},i})$
$\mathbb{R}^n$
$\begin{bmatrix} \widehat{x}_{\text{ls}} \\ -1 \end{bmatrix}$

Note that the former geometric interpretation is in the space $\mathbb{R}^m$, while the latter is in the (data space) $\mathbb{R}^{n+1}$.

Exercise problem 52. [Derivation of solution x_{ln} via Lagrange multipliers] Assuming that $m \geq n = \text{rank}(A)$, *i.e.*, A is full column rank, show that

$$\widehat{x}_{\text{ls}} = (A^\top A)^{-1} A^\top y.$$

□

Notes:

- $A_{\text{ls}} := (A^\top A)^{-1} A^\top$ is a left-inverse of A
- $\widehat{x}_{\text{ls}}$ is a linear function of y (given by the matrix A_{ls})
- If A is square, $\widehat{x}_{\text{ls}} = A^{-1} y$ (*i.e.*, $A_{\text{ls}} = A^{-1}$)
- $\widehat{x}_{\text{ls}}$ is an exact solution if $Ax = y$ has an exact solution
- $\widehat{y}_{\text{ls}} := A\widehat{x}_{\text{ls}} = A(A^\top A)^{-1} A^\top y$ is a least-squares approximation of y

Projector onto the image of A and orthogonality principle

The $m \times m$ matrix

$$\Pi_{\text{image}(A)} := A(A^\top A)^{-1}A^\top$$

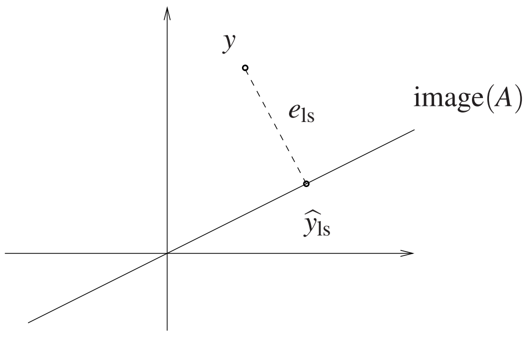
is the orthogonal projector onto the subspace $\mathscr{L} := \text{image}(A)$. Suppose that the columns of A form an orthonormal basis for $\mathscr{L}$. Then, recall that $\Pi_{\text{image}(Q)} := AA^\top$.

The least-squares residual vector

$$e_{\text{ls}} := y - A\widehat{x}_{\text{ls}} = \underbrace{\left(I_m - A(A^\top A)^{-1}A^\top\right)}_{\Pi_{(\text{image}(A))^\perp}} y$$

is orthogonal to image(A)

$$\langle e_{\text{ls}}, A\widehat{x}_{\text{ls}}\rangle = y^\top \left(I_m - A(A^\top A)^{-1}A^\top\right)A\widehat{x}_{\text{ls}} = 0. \tag{3.2}$$



Exercise problem 53. Show that the orthogonality condition (3.2) is a necessary and sufficient condition for $\widehat{x}_{\text{ls}}$ being a least squares approximate solution to $Ax = b$.

□

Least-squares via QR factorization

Let $A = QR$ be the QR factorization of A. We have,

$$\begin{aligned}(A^\top A)^{-1}A^\top &= (R^\top Q^\top QR)^{-1}R^\top Q^\top \\ &= (R^\top Q^\top QR)^{-1}R^\top Q^\top = R^{-1}Q^\top,\end{aligned}$$

so that

$$\widehat{x}_{\text{ls}} = R^{-1}Q^\top y \quad \text{and} \quad \widehat{y}_{\text{ls}} := Ax_{\text{ls}} = QQ^\top y.$$

Exercise problem 54 (Least-squares with an increasing number of columns in A). Let $A =: \begin{bmatrix} a_1 & \cdots & a_n \end{bmatrix}$ and consider the sequence of least squares problems

$$A^i x^i = y, \qquad \text{where } A^i := \begin{bmatrix} a_1 & \cdots & a_i \end{bmatrix}, \quad \text{for } i = 1, \ldots, n$$

Define R_i as the leading $i \times i$ submatrix of R and let $Q_i := \begin{bmatrix} q_1 & \cdots & q_i \end{bmatrix}$. Show that

$$\widehat{x}^i_{\text{ls}} = R_i^{-1}Q_i^\top y.$$

□

Weighted least-squares

Given a positive definite matrix $W \in \mathbb{R}^{m\times m}$, define the wighted 2-norm

$$\|e\|_W^2 := e^\top We.$$

and the weighted least-squares approximate solution

$$\widehat{x}_{W,\mathrm{ls}} := \arg\min_x \|y - Ax\|_W^2.$$

Exercise problem 55. Show that

$$\widehat{x}_{W,\mathrm{ls}} = (A^\top WA)^{-1}A^\top Wy,$$

and that the least-squares orthogonality principle holds for the weighted least-squares problem as well by replacing the inner product $\langle e, y\rangle$ by the weighted inner product

$$\langle e, y\rangle_W := e^\top Wy.$$

□

Recursive least-squares

The least-squares criterion is

$$\|y - Ax\|_2^2 = \sum_{i=1}^m (y_i - a_i^\top x)^2$$

where $a_i^\top$ is the ith row of A. We consider the sequence of least-squares problems

$$\text{minimize} \quad \sum_{i=1}^k (y_i - a_i^\top x)^2$$

the solutions of which are

$$\widehat{x}_{\mathrm{ls}}(k) := \left(\sum_{i=1}^k a_i a_i^\top\right)^{-1} \sum_{i=1}^m a_i y_i.$$

The meaning is that the measurements (a_i, y_i) come sequentially (in time) and we aim to compute a solution each time a new data point arrives. Instead of recomputing the solution from scratch, we can recursively update $\widehat{x}_{\mathrm{ls}}(k-1)$ in order to obtain $\widehat{x}_{\mathrm{ls}}(k)$.

Recursive algorithm

- Initialization: $P(0) = 0 \in \mathbb{R}^{n\times n}$, $q(0) = 0 \in \mathbb{R}^n$
- For $m = 0, 1, \ldots, m$
- $P(k+1) := P(k) + a_{k+1}a_{k+1}^\top$, $q(k+1) := q(k) + a_{k+1}y_{k+1}$
- If $P(k)$ is invertible, $\widehat{x}_{\mathrm{ls}}(k) = P^{-1}(k)q(k)$.

On each step, the algorithm requires inversion of an $n \times n$ matrix, which requires $O(n^3)$ operations. At certain k, $P(k)$ being invertible implies that $P(k')$ is invertible, for all $k' > k$.

The computational complexity of the algorithm can be decreased to $O(n^2)$ operations per step by using the following result about the inverse of matrix with rank-1 update

$$(P + aa^\top)^{-1} = P^{-1} - \frac{1}{1 + a^\top P^{-1}a}(P^{-1}a)(P^{-1}a)^\top.$$

Multiobjective least-squares

Least-squares minimizes the cost function

$$J_1(x) := \|Ax - y\|_2^2.$$

Consider a second cost function

$$J_2(x) := \|Bx - z\|_2^2,$$

which we want to minimize together with J_1. Usually the criteria $\min_x J_1(x)$ and $\min_x J_2(x)$ are competing. A common example is $J_2(x) := \|x\|_2^2$ — minimize J_1 with small x.

The set of achievable objectives is

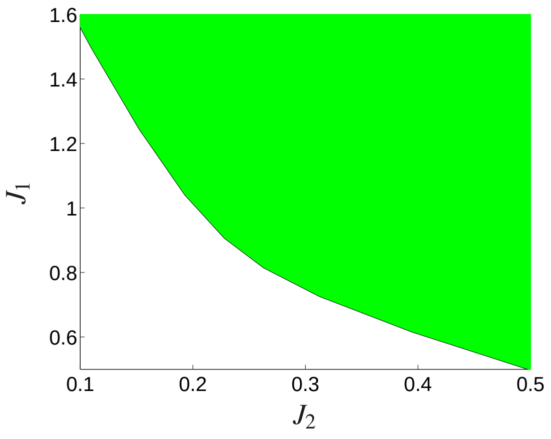
$$\{\,(\alpha,\beta) \in \mathbb{R}^2 \mid \exists\, x \in \mathbb{R}^n \text{ subject to } J_1(x) = \alpha,\ J_2(x) = \beta\,\}$$

Its boundary is the optimal trade-off curve and the corresponding x's are called *Pareto optimal*.

A common method for "solving" multiobjective optimization problems is secularization. For any $\mu \geq 0$, the problem

$$\widehat{x}(\mu) = \arg\min_x J_1(x) + \mu J_2(x)$$

produces a Pareto optimal point. For a convex problem (such as the the multiobjective least-squares), by varying $\mu \in [0,\infty)$, $\widehat{x}(\mu)$ sweeps all Pareto optimal solutions.



Regularized least-squares

Exercise problem 56. Show that the solution of the *Tychonov regularization* problem

$$\widehat{x}_{\text{reg}} = \arg\min_x \|Ax - b\|_2^2 + \mu\|x\|_2^2$$

is

$$\widehat{x}_{\text{reg}} = (A^\top A + \mu I_n)^{-1} A^\top y.$$

□

Note that $\widehat{x}_{\text{reg}}$ exists for any $\mu > 0$, independent on size and rank of A. The parameter μ controls the trade-off between

- fitting accuracy $\|Ax - b\|_2$, and
- solution size $\|x\|_2$.

For small μ, the solution is larger but gives better fit. For large μ, the solution is smaller but the fit is worse. In the extreme case $\mu = 0$, assuming that the system $Ax = b$ is overdetermined, the regularized least-squares problem is equivalent to the standard least-squares problem, which does not constrain the size of x. In the other extreme $\mu \to 0$, assuming that $Ax = b$ is underdetermined, the regularized least-squares problem tends to the least-norm problem.

3.2 Least-norm

Consider an underdetermined system $Ax = y$, with full rank $A \in \mathbb{R}^{m\times n}$. The set of solutions is

$$\mathscr{A} := \{\, x \in \mathbb{R}^n \mid Ax = y \,\} = \{\, x_\text{p} + z \mid z \in \ker(A) \,\} = x_\text{p} + \ker(A).$$

where x_p is a particular solution, *i.e.*, $Ax_\text{p} = y$. The least-norm solution is defined by the optimization problem

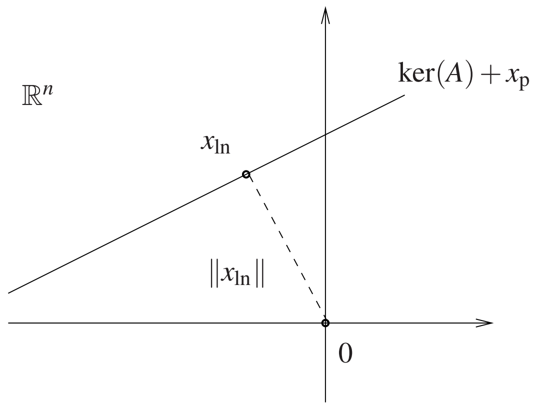
$$x_\text{ln}^2 := \arg\min_x \|x\|_2 \quad \text{subject to} \quad Ax = y. \tag{3.3}$$

Exercise problem 57 (Derivation of solution x_ln via Lagrange multipliers). Assuming that $n \geq m = \operatorname{rank}(A)$, *i.e.*, A is full row rank, show that

$$x_\text{ln} = A^\top (AA^\top)^{-1} y.$$

□

A geometric interpretation of (1.4) is the projection of 0 onto the solution set $\mathscr{A}$.



Exercise problem 58. The orthogonality principle for least-norm is $x_\text{ln} \perp \ker(A)$. Show that it is a necessary and sufficient condition for optimality of x_ln

□

Let $A^\top = QR$ be the QR factorization of $A^\top$. The right inverse of A is

$$A^\top (AA^\top)^{-1} = QR(R^\top Q^\top QR)^{-1} = Q(R^\top)^{-1},$$

so that

$$x_\text{ln} = Q(R^\top)^{-1} y.$$

3.3 Total least-squares

The least-squares method minimizes the 2-norm of the equation error $e(x) := y - Ax$

$$\min_{x,e} \|e\|_2 \quad \text{subject to} \quad Ax = y - e$$

Alternatively, the equation error e can be viewed as a correction on y. The total least-squares method is motivated by the asymmetry of the least-squares method: both A and b are given data, but only b is corrected. The total least squares problem is defined by the optimization problem

$$\text{minimize}_{x,\widetilde{A},\widetilde{y}} \quad \big\| \big[\widetilde{A} \;\; \widetilde{y}\big] \big\|_\text{F} \quad \text{subject to} \quad (A + \widetilde{A})x = y + \widetilde{y}$$

Here $\widetilde{A}$ is the correction on A and $\widetilde{y}$ is the correction on y. The Frobenius norm $\|C\|_\text{F}$ of $C \in \mathbb{R}^{m\times n}$ is defined as

$$\|C\|_\text{F} := \sqrt{\sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij}^2}.$$

Geometric interpretation of the total least squares criterion

In the case $n = 1$, the problem of solving approximately $Ax = y$ is

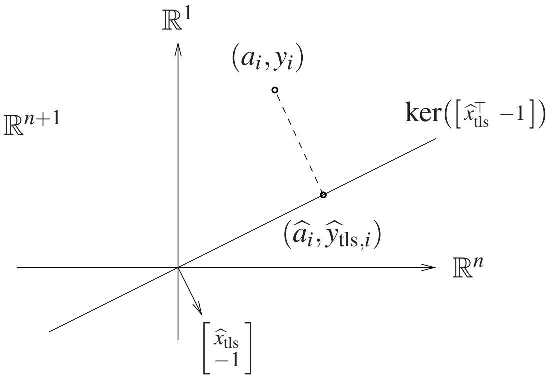
$$\begin{bmatrix} a_1 \\ \vdots \\ a_m \end{bmatrix} x = \begin{bmatrix} y_1 \\ \vdots \\ y_m \end{bmatrix}, \qquad \text{where} \quad x \in \mathbb{R}. \tag{3.4}$$

A geometric interpretation of the total least squares problem (3.4) is: fit a line

$$\mathscr{L}(x) := \{\, (a,b) \mid ax = b \,\}$$

passing through the origin to the points $(a_1, y_1), \dots, (a_m, y_m)$.

- least squares minimizes sum of squared *vertical* distances from (a_i, y_i) to $\mathscr{L}(x)$,
- total least squares minimizes sum of squared *orthogonal* distances from (a_i, y_i) to $\mathscr{L}(x)$.


$\mathbb{R}^1$
(a_i, y_i)
$\mathbb{R}^{n+1}$
$\ker([\widehat{x}_{\text{tls}}^\top \; -1])$
$(\widehat{a}_i, \widehat{y}_{\text{tls},i})$
$\mathbb{R}^n$
$\begin{bmatrix} \widehat{x}_{\text{tls}} \\ -1 \end{bmatrix}$

Solution of the total least squares problem

Theorem 59. *Let $\begin{bmatrix} A & y \end{bmatrix} = U \Sigma V^\top$ be the SVD of the data matrix $\begin{bmatrix} A & y \end{bmatrix}$ and*

$$\Sigma := \operatorname{diag}(\sigma_1, \dots, \sigma_{n+1}), \quad U := \begin{bmatrix} u_1 & \cdots & u_{n+1} \end{bmatrix}, \quad V := \begin{bmatrix} v_1 & \cdots & v_{n+1} \end{bmatrix}.$$

A total least squares solution exists if and only if $v_{n+1,n+1} \neq 0$ (last element of v_{n+1}) and is unique if and only if $\sigma_n \neq \sigma_{n+1}$.

In the case when a total least squares solution exists and is unique, it is given by

$$\widehat{x}_{\text{tls}} = -\frac{1}{v_{n+1,n+1}} \begin{bmatrix} v_{1,n+1} \\ \vdots \\ v_{n,n+1} \end{bmatrix}$$

and the corresponding total least squares corrections are

$$\begin{bmatrix} \widetilde{A}_{\text{tls}} & \widetilde{y}_{\text{tls}} \end{bmatrix} = -\sigma_{n+1} u_{n+1} v_{n+1}^\top.$$

3.4 Low-rank approximation

The low-rank approximation problem is defined as: Given a matrix $A \in \mathbb{R}^{m \times n}$, $m \geq n$, and an integer r, $0 < r < n$, find

$$\widehat{A}^* := \arg\min_{\widehat{A}} \|A - \widehat{A}\| \quad \text{subject to} \quad \operatorname{rank}(\widehat{A}) \leq r. \tag{3.5}$$

$\widehat{A}^*$ is an optimal rank-r approximation of A with respect to the norm $\|\cdot\|$, *e.g.*,

$$\|A\|_{\text{F}}^2 := \sum_{i=1}^{m} \sum_{j=1}^{n} a_{ij}^2 \qquad \text{or} \qquad \|A\|_2 := \max_x \frac{\|Ax\|_2}{\|x\|_2}$$

Theorem 60 (Solution via SVD)**.** *Let $A = U\Sigma V^\top$ be the SVD of A and define*

$$U =: \begin{array}{cc} r & r-n \\ \begin{bmatrix} U_1 & U_2 \end{bmatrix} \end{array}\; n\,, \quad \Sigma =: \begin{array}{ccc} r & r-n & \\ \Sigma_1 & 0 & r \\ 0 & \Sigma_2 & r-n \end{array} \quad \text{and} \quad V =: \begin{array}{cc} r & r-n \\ \begin{bmatrix} V_1 & V_2 \end{bmatrix} \end{array}\; n\,.$$

An solution to (3.5) is

$$\widehat{A}^* = U_1\Sigma_1V_1^\top.$$

It is unique if and only if $\sigma_r \neq \sigma_{r+1}$.

3.5 Notes and references

Least-squares and least-norm are standard topics in both numerical linear algebra and engineering. Numerical aspects of the problem are considered in [Bjö96]. For an overview of total least squares problem, see [MV07]

Bibliography

[Bjö96] Å. Björck. *Numerical Methods for Least Squares Problems.* SIAM, 1996.

[MV07] I. Markovsky and S. Van Huffel. Overview of total least squares methods. *Signal Processing*, 87:2283–2302, 2007.