Linear Algebra (Applications)

Mark DiBattista

May 27, 2019

Abstract

(Real) Linear Algebra is the study of vectors and linear vector operators whose elements are real numbers and whose rules of combination are governed by vector addition and scalar multiplication. Vectors are organized into collections, called vector spaces, and operators can be defined by induced partitions, as one vector space is mapped to another. In fact the full set of n-dimensional vectors – represented as lists of n real numbers – is partitioned by linear operators into no more than n vector subspaces, and the behavior of operators within these subspaces is classified into just a handful of distinct categories.

1 Suggested Resource Materials

Useful source texts:

- Basic linear algebra: *Applied Linear Algebra*, Noble & Daniel
  Introduction to Linear Algebra, Strang
- Computational perspective: *Matrix Computations*, Golub
- Optimization applications: *Practical Optimization*, Gill, Murray & Wright
- Many, many matrix formulas: (http://www.math.uwaterloo.ca/ hwolkowi/matrixcookbook.pdf)

Throughout the text the acronym, *LAN*, refers to the companion writeup, *Linear Algebra Notes* and *PN* refers to *Probability Notes*, in which information is referenced by chapter and/or numbered equation.

2 Real Functions and Vector Operators

2.1 Conic Functions

Two-dimensional scalar conic functions can be expressed as the level curves of the quadratic form (*cf.* *LAN*, §3.8),

$$f(\mathbf{x}) = \mathbf{x}^\top A\mathbf{x}, \tag{1}$$

for which the matrix, A, is self-adjoint: $A = A^\top$. The classification of conic functions – ellipse, hyperbola or parabola – depends on the signs of the eigenvalues of the matrix, A.

The eigensystems of self-adjoint matrices are given by the Spectral Theorem (*cf. LAN*, §4.2): all eigenvalues are real, and all associated eigenvectors form an orthonormal set. Diagonalizing the matrix we have

$$A = VDV^\top \Rightarrow \begin{cases} \text{eigenvalues: } D \text{ is diagonal, } D = \begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix} \\ \text{eigenvectors: } V \text{ is orthogonal, } VV^\top = V^\top V = I \end{cases} \tag{2}$$

Introducing a change of basis (*cf. LAN*, §3.9.3) to align with coordinate system that coincides with the eigenvectors:

$$\mathbf{x}' = V^\top \mathbf{x} \Rightarrow \begin{cases} \mathbf{x}^\top A \mathbf{x} & = \mathbf{x}^\top V D V^\top \mathbf{x} \\ & = \left(V^\top \mathbf{x}\right)^\top D \left(V^\top \mathbf{x}\right) \\ & = \mathbf{x}'^\top D \mathbf{x}' \end{cases} \tag{3}$$

the conic equations take the form,

$$f(x', y') = \lambda_1 x'^2 + \lambda_2 y'^2 \Rightarrow \begin{cases} \lambda_1, \lambda_2 \text{ same sign}, & \text{ellipse} \\ \lambda_1, \lambda_2 \text{ different sign}, & \text{hyperbola} \\ \lambda_1 \text{ or } \lambda_2 = 0, & \text{parabola} \end{cases} \tag{4}$$

In general if the matrix, A, is an $n \times n$ positive- or negative-definite matrix, then the level surfaces defined by the equation in (1) take the form of nested ellipsoids.

2.1.1 Link between the Discriminant and the Characteristic Equation

Filling in the entries in the 2-dimensional matrix, A, leads to the standard conic equation in basic algebra:

$$\left.\begin{aligned} A &= \begin{pmatrix} a & \frac{1}{2}b \\ \frac{1}{2}b & c \end{pmatrix} \\ \mathbf{x} &= \begin{pmatrix} x \\ y \end{pmatrix} \end{aligned}\right\} \Rightarrow \mathbf{x}^\top A \mathbf{x} = ax^2 + bxy + cy^2. \tag{5}$$

The characteristic equation (*cf. LAN*, §4, (83)) of the matrix,

$$\det(A - \lambda I) = \begin{vmatrix} a - \lambda & \frac{1}{2}b \\ \frac{1}{2}b & c - \lambda \end{vmatrix} = \lambda^2 - (a + c)\lambda - \frac{1}{4}(b^2 - 4ac) = 0, \tag{6}$$

has the solution

$$\lambda = \frac{(a + c) \pm \sqrt{(a + c)^2 + (b^2 - 4ac)}}{2} = \frac{a + c}{2}\left(1 \pm \sqrt{1 + \frac{b^2 - 4ac}{(a + c)^2}}\right). \tag{7}$$

The relative signs of the eigenvalues are determined by the sign of the **discriminant**,

$$b^2 - 4ac \Rightarrow \begin{cases} > 0, & \text{ellipse} \\ = 0, & \text{parabola} \\ < 0, & \text{hyperbola} \end{cases} \tag{8}$$

which match the conditions (4). Also, note that the characteristic equation can be expressed in terms of the trace and determinant of the matrix:

$$\lambda = \frac{\operatorname{tr} A}{2}\left(1 \pm \sqrt{1 + \frac{\det A}{\operatorname{tr}^2 A}}\right), \tag{9}$$

both of which are invariant under orthogonal transformation of the matrix.

2.2 Differential Vector Operators

2.2.1 Gradient and Hessian

Multidimensional functions, $f : \mathbb{R}^n \to \mathbb{R}$, map vectors in the n-dimensional domain to a scalar function. The directional derivatives of multidimensional functions, for which the derivatives are taken in directions that coincide with the natural basis of the vector space, can be ordered in a vector form that contains the complete information necessary to calculate the rate of change of the function in arbitrary directions of the space. The **gradient operator** for the function, f, takes the form

$$\text{Gradient: } \nabla \equiv \begin{pmatrix} \frac{\partial}{\partial x_1} \\ \vdots \\ \frac{\partial}{\partial x_n} \end{pmatrix} \Rightarrow \nabla f(\mathbf{x}) \equiv \begin{pmatrix} \frac{\partial f}{\partial x_1} \\ \vdots \\ \frac{\partial f}{\partial x_n} \end{pmatrix}, \tag{10}$$

and the directional derivative at the arbitrary point, $\mathbf{x}$, in the arbitrary direction, $\mathbf{s}$, is expressed as the inner product,

$$\langle \mathbf{s}, \nabla f(\mathbf{x}) \rangle \equiv \mathbf{s}^\top \nabla f(\mathbf{x}) \equiv \nabla f(\mathbf{x})^\top \mathbf{s} \equiv \sum_{i=1}^{n} s_i \frac{\partial f}{\partial x_i}. \tag{11}$$

Similarly the **Hessian operator** for the function, f, records the second-order directional derivatives, again in directions aligned with the natural basis vectors, taken pairwise in matrix form ,

$$\text{Hessian: } H \equiv \begin{pmatrix} \frac{\partial^2}{\partial x_1^2} & \cdots & \frac{\partial^2}{\partial x_1 \partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial^2}{\partial x_m \partial x_1} & \cdots & \frac{\partial^2}{\partial x_m \partial x_n} \end{pmatrix} \Rightarrow Hf(\mathbf{x}) \equiv \begin{pmatrix} \frac{\partial^2 f}{\partial x_1^2} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_m \partial x_1} & \cdots & \frac{\partial^2 f}{\partial x_m \partial x_n} \end{pmatrix}. \tag{12}$$

The curvature of the surface at the arbitrary point, $\mathbf{x}$, in the arbitrary direction, $\mathbf{s}$, is given by the quadratic form,

$$\langle \mathbf{s}, Hf(\mathbf{x})\mathbf{s} \rangle \equiv \mathbf{s}^\top Hf(\mathbf{x})\mathbf{s}. \tag{13}$$

In the special cases for which the function, f, is itself an inner product or quadratic form the gradient is a function of the underlying vector,

$$\text{inner product : } f(\mathbf{x}) = \langle \mathbf{x}, \mathbf{c} \rangle \Rightarrow \nabla_{\mathbf{x}} \left(\mathbf{x}^\top \mathbf{c}\right) = \nabla_{\mathbf{x}} \left(\mathbf{c}^\top \mathbf{x}\right) = \mathbf{c}; \tag{14}$$

$$\text{quadratic form: } f(\mathbf{x}) = \frac{1}{2}\mathbf{x}^\top C\mathbf{x} \Rightarrow \nabla_{\mathbf{x}} \left(\frac{1}{2}\mathbf{x}^\top C\mathbf{x}\right) = C\mathbf{x}. \tag{15}$$

Here, the subscripts indicate the variables against which the derivatives are applied.

2.2.2 Taylor's Approximations in Multidimensional Spaces

Taylor's Theorem links information on a function at a single point – the value of the function and derivatives at all orders at the point – to the value of the function at arbitrary points at arbitrary distances. If we retain only partial information locally – only the value at the point and the first and second derivatives – we obtain an approximation for the value of the function at arbitrary points, an approximation that degrades with the distance of the arbitrary point.

Choosing the local point as $\mathbf{x}$, the second-order **Taylor's approximation for scalar functions**, $f : \mathbb{R}^n \to \mathbb{R}$, evaluated at the arbitrary point, $\mathbf{x} + \Delta\mathbf{x}$, depends only on the gradient and Hessian of the function,

$$f(\mathbf{x} + \Delta\mathbf{x}) \approx f(\mathbf{x}) + \nabla f(\mathbf{x})^\top \Delta\mathbf{x} + \frac{1}{2}\Delta\mathbf{x}^\top H f(\mathbf{x}) \Delta\mathbf{x}. \tag{16}$$

Furthermore, **Taylor's Theorem for multidimensional functions**, $\mathbf{f} : \mathbb{R}^n \to \mathbb{R}^m$ leads to the vector representation for which the gradient and Hessian operators are n-dimensional, and the result is stored in the m-dimensional range space,

$$\mathbf{f}(\mathbf{x} + \Delta\mathbf{x}) \equiv \begin{pmatrix} f_1(\mathbf{x} + \Delta\mathbf{x}) \\ \vdots \\ f_m(\mathbf{x} + \Delta\mathbf{x}) \end{pmatrix} \approx \begin{pmatrix} f_1(\mathbf{x}) + \nabla f_1(\mathbf{x})^\top \Delta\mathbf{x} + \frac{1}{2}\Delta\mathbf{x}^\top H f_1 \Delta\mathbf{x} \\ \vdots \\ f_m(\mathbf{x}) + \nabla f_m(\mathbf{x})^\top \Delta\mathbf{x} + \frac{1}{2}\Delta\mathbf{x}^\top H f_m \Delta\mathbf{x} \end{pmatrix}. \tag{17}$$

The equations in (16) and (17) can be interpreted as second-order algebraic approximations to the true function, f. Frequently, this amounts to replacing the complicated surface with elliptic paraboloids or hyperboloids whose value, gradient and Hessian match the function at the point, $\mathbf{x}$. Numerical methods in optimization generate extrema in non-linear functions, f, by calculating the extrema of a series second-order approximations, and iterating until convergence.

2.2.3 Jacobian and Differential Volume

The volume of an n-dimensional parallelepiped whose vertex is enclosed by the vectors, $\mathbf{c}_1, \ldots, \mathbf{c}_n$, is given by, (*cf. LAN*, §3.7, (56)),

$$A = \begin{pmatrix} \mathbf{c}_1 & \cdots & \mathbf{c}_n \end{pmatrix} \Rightarrow \text{volume: } |\det A| = \left| \bigwedge_{i=1}^{n} \mathbf{c}_i \right|. \tag{18}$$

The Cartesian **differential volume element** for an n-dimensional space can be represented as the wedge product of the differentials associated with each direction,

$$dV = dx_1 \wedge \cdots \wedge dx_n. \tag{19}$$

If we introduce a change of coordinate system, $(x_1, \ldots, x_n) \to (x'_1, \ldots, x'_n)$, then the **Jacobian matrix**,

$$J = \begin{pmatrix} \frac{\partial x_1}{\partial x'_1} & \cdots & \frac{\partial x_1}{\partial x'_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial x_m}{\partial x'_1} & \cdots & \frac{\partial x_m}{\partial x'_n} \end{pmatrix}, \tag{20}$$

gives the 'volume element' of the transformation. From the transformation of (total) differentials we can construct the transformed volume element,

$$dx_i = \sum_{j=1}^{n} \frac{\partial x_i}{\partial x'_j} dx'_j \Rightarrow \bigwedge_{i=1}^{n} dx_i = |J| \bigwedge_{j=1}^{n} dx'_j \Rightarrow dV = |J| dV', \tag{21}$$

whose scaling is given by the determinant of the Jacobian matrix.

Notica also that the Hessian matrix in (12) can be expressed as the Jacobian of the gradient of the transformation.

3 Least Squares

Given an $m \times n$ data matrix, A, and an $m \times 1$ vector, $\mathbf{b}$, it is a common optimization problem to find the $n \times 1$ vector, $\mathbf{x}$ that minimizes the distance between the linear transformation, $A\mathbf{x}$, and the target vector, $\mathbf{b}$. For the case in which $m = n$, and the matrix is square, the possible solutions to the equation,

$$A\mathbf{x} - \mathbf{b} = 0, \tag{22}$$

are governed by the Fredholm Alternative (*cf. LAN*, §7.1), with no solution, a unique solution, or an infinity of solutions depending on the rank and the relative extent of the image of the matrix. For the **overdetermined problem**, $m > n$, the probability that the target vector, $\mathbf{b}$, lies in the column space of A is remote, and optimization determines the vector, $\mathbf{x}$, that minimizes the distance between transformed and target vectors. If the 'distance' is interpreted as the vector 2-norm, we have the **least-squares problem**,

$$\min_{\mathbf{x}} ||A\mathbf{x} - \mathbf{b}||_2^2 \Rightarrow \nabla_{\mathbf{x}} \left(A\mathbf{x} - \mathbf{b}\right)^\top \left(A\mathbf{x} - \mathbf{b}\right) = 0, \tag{23}$$

which expresses the distance-minimizing solution as the vanishing gradient of an inner product.

3.1 Generalized Inverse Solution

Expanding the inner product in (23) and applying the gradient operator, defined in (14), term-by-term yields,

$$\begin{aligned}
\nabla_{\mathbf{x}} \left(A\mathbf{x} - \mathbf{b}\right)^\top \left(A\mathbf{x} - \mathbf{b}\right) &= \nabla_{\mathbf{x}} \left(A\mathbf{x}\right)^\top \left(A\mathbf{x}\right) - \nabla_{\mathbf{x}} \left(A\mathbf{x}\right)^\top \mathbf{b} - \nabla_{\mathbf{x}} \mathbf{b}^\top A\mathbf{x} + \nabla_{\mathbf{x}} \mathbf{b}^\top \mathbf{b} && (24)\\
&= \nabla_{\mathbf{x}} \mathbf{x}^\top A^\top A\mathbf{x} - \nabla_{\mathbf{x}} \mathbf{x}^\top A^\top \mathbf{b} - \nabla_{\mathbf{x}} \mathbf{b}^\top A\mathbf{x} + \nabla_{\mathbf{x}} \mathbf{b}^\top \mathbf{b} && (25)\\
&= 2A^\top A\mathbf{x} - 2A^\top \mathbf{b} = 0 && (26)\\
\Rightarrow \mathbf{x}_{LS} &= \left(A^\top A\right)^{-1} A^\top \mathbf{b}. && (27)
\end{aligned}$$

Here, the least-squares solution is achieved by applying the **generalized inverse** operator, $\left(A^\top A\right)^{-1} A^\top$, to the target vector, an operation that requires the data matrix to be of full rank, $\text{rank}(A) = n$. Note that the action of the matrix, A, on the least-square solution, $\mathbf{x}_{LS}$, is equivalent to projecting the target vector onto the column space of A,

$$A\mathbf{x}_{LS} = A\left(A^\top A\right)^{-1} A^\top \mathbf{b} \equiv P_A \mathbf{b} \tag{28}$$

See the discussion in *LAN*, §3.10 for properties of projection operators.

3.2 Iterative Solution (Bayesian Form)

The solution to the least-squares problem in (24) requires *all* information in the data matrix to generate generalized inverse. If, for example, data is gathered at two separate time points, or if data is arriving continuously, generalized inverses are created from ever-larger data matrices that require increasing numbers of computations. Since the data points that make up the matrix are independent and all operations in the creation of the generalized inverse are linear, it is possible to derive an iterative solution that build upon prior information, one that can be cast into Bayesian terms (cf. *PN*, §11. The two methods generate identical solutions – the iterative method is simply a more efficient implementation.

Let A_i be an $m \times n$ matrix that represents m prior data measurements (here, $i = m$) taken from an n-dimensional space with known inverse covariance matrix, $\left(A_i^\top A_i\right)^{-1}$, and let the vector, $\mathbf{b}_i$, be the associated prior target vector of responses. A single additional datapoint, $\mathbf{a}_{i+1}$, and response, b_{i+1}, can be represented in block form as

$$A_{i+1} = \begin{pmatrix} A_i \\ \mathbf{a}_{i+1}^{-1} \end{pmatrix} \tag{29}$$

$$\mathbf{b}_{i+1} = \begin{pmatrix} \mathbf{b}_i \\ b_{i+1} \end{pmatrix}. \tag{30}$$

The new inverse scatter matrix can also be represented in block form as

$$\left(A_{i+1}^\top A_{i+1}\right)^\top = \left(A_i^\top A_i + \mathbf{a}_{i+1}\mathbf{a}_{i+1}^\top\right)^{-1}, \tag{31}$$

whose expansion can be derived from the the Woodbury formula, *cf. LAN*, §6.3,

$$\left(A - BD^{-1}C\right)^{-1} = (I + A^{-1}B\left(D - CA^{-1}B\right)^{-1}C)A^{-1}. \tag{32}$$

Indentifying the blocks in (32) with the elements in the updated inverse scatter matrix in (37),

$$A \to A_i^\top A_i \tag{33}$$

$$B \to \mathbf{a}_{i+1} \tag{34}$$

$$C \to \mathbf{a}_{i+1}^\top \tag{35}$$

$$D \to -1 \tag{36}$$

yields the expansion,

$$\begin{aligned} \left(A_{i+1}^\top A_{i+1}\right)^{-1} &= \left(I - \frac{\left(A_i^\top A_i\right)^{-1}\mathbf{a}_{i+1}\mathbf{a}_{i+1}^\top}{1 + \mathbf{a}_{i+1}^\top\left(A_i^\top A_i\right)^{-1}\mathbf{a}_{i+1}}\right)\left(A_i^\top A_i\right)^{-1} \\ &= \left(I - \mathbf{k}_{i+1}\mathbf{a}_{i+1}^\top\right)\left(A_i^\top A_i\right)^{-1}, \end{aligned} \tag{37}$$

in which the updated **gain matrix**,

$$\mathbf{k}_{i+1} \equiv \frac{\left(A_i^\top A_i\right)^{-1}\mathbf{a}_{i+1}}{1 + \mathbf{a}_{i+1}^\top\left(A_i^\top A_i\right)^{-1}\mathbf{a}_{i+1}} \tag{38}$$

depends only of the prior inverse scatter matrix, $(A_i^\top A_i)^{-1}$, and the new data point, $\mathbf{a}_{i+1}$. Introducing the expansion in (37) into the least-squares solution in (24) yields

$$\begin{aligned}
\mathbf{x}_{i+1} &= \left(A_{i+1}^\top A_{i+1}\right)^{-1} A_{i+1}^\top \mathbf{b}_{i+1} \\
&= \left(I - \mathbf{k}_{i+1}\mathbf{a}_{i+1}^\top\right)\left(A_i^\top A_i\right)^{-1}\begin{pmatrix} A_i^\top & \mathbf{a}_{i+1}\end{pmatrix}\begin{pmatrix}\mathbf{b}_i \\ b_{i+1}\end{pmatrix} \\
&= \left(I - \mathbf{k}_{i+1}\mathbf{a}_{i+1}^\top\right)\left(A_i^\top A_i\right)^{-1}\left(A_i^\top \mathbf{b}_i + \mathbf{a}_{i+1} b_{i+1}\right) \\
&= \left(I - \mathbf{k}_{i+1}\mathbf{a}_{i+1}^\top\right)\mathbf{x}_i + \mathbf{k}_{i+1} b_{i+1}. \qquad (39)
\end{aligned}$$

Here, the updated solution, $\mathbf{x}_{i+1}$, is Bayesian in form for which the gain matrix, $\mathbf{k}_{i+1}$, determines the weighting between the prior estimate, $\mathbf{x}_i$, and the new information contained in the measurements, $\mathbf{a}_{i+1}$ and $\mathbf{b}_{i+1}$.