Bonus Lecture: Intro/refresher in Matrix Algebra

UNE course:
The search for selection
3-- 7 Feb 2020

Bruce Walsh (University of Arizona)
jbwalsh@email.arizona.edu

Topics

- Definitions, dimensionality, addition, subtraction
- Matrix multiplication
- Inverses, solving systems of equations
- Quadratic products and covariances
- The multivariate normal distribution
- Eigenstructure
- Basic matrix calculations in R

Matrices: An array of elements

Vectors: A matrix with either one row or one column.

Usually written in bold lowercase, e.g. **a, b, c**

$$\mathbf{a} = \begin{pmatrix} 12 \\ 13 \\ 47 \end{pmatrix} \quad \mathbf{b} = \begin{pmatrix} 2 & 0 & 5 & 21 \end{pmatrix}$$

Column vector (3 x 1)

Row vector (1 x 4)

Dimensionality of a matrix: r x c (rows x columns)
think of Railroad Car

General Matrices

Usually written in bold uppercase, e.g. **A, C, D**

$$\mathbf{C} = \begin{pmatrix} 3 & 1 & 2 \\ 2 & 5 & 4 \\ 1 & 1 & 2 \end{pmatrix} \quad \mathbf{D} = \begin{pmatrix} 0 & 1 \\ 3 & 4 \\ 2 & 9 \end{pmatrix}$$

(3 x 3)

Square matrix

(3 x 2)

Dimensionality of a matrix: r x c (rows x columns)
think of Railroad Car

A matrix is defined by a list of its elements.
B has ij-th element B_{ij} -- the element in row i and column j

Addition and Subtraction of Matrices

If two matrices have the same dimension (both are r x c), then matrix addition and subtraction simply follows by adding (or subtracting) on an element by element basis

Matrix addition: $(A+B)_{ij} = A_{ij} + B_{ij}$

Matrix subtraction: $(A-B)_{ij} = A_{ij} - B_{ij}$

Examples:

$$\mathbf{A} = \begin{pmatrix} 3 & 0 \\ 1 & 2 \end{pmatrix} \quad \text{and} \quad \mathbf{B} = \begin{pmatrix} 1 & 2 \\ 2 & 1 \end{pmatrix}$$

$$\mathbf{C} = \mathbf{A} + \mathbf{B} = \begin{pmatrix} 4 & 2 \\ 3 & 3 \end{pmatrix} \quad \text{and} \quad \text{D = A-B} = \begin{pmatrix} 2 & -2 \\ -1 & 1 \end{pmatrix}$$

Partitioned Matrices

It will often prove useful to divide (or partition) the elements of a matrix into a matrix whose elements are itself matrices.

$$\mathbf{C} = \begin{pmatrix} 3 & 1 & 2 \\ 2 & 5 & 4 \\ 1 & 1 & 2 \end{pmatrix} = \begin{pmatrix} 3 & \vdots & 1 & 2 \\ \cdots & \cdots & \cdots & \cdots \\ 2 & \vdots & 5 & 4 \\ 1 & \vdots & 1 & 2 \end{pmatrix} = \begin{pmatrix} \mathbf{a} & \mathbf{b} \\ \mathbf{d} & \mathbf{B} \end{pmatrix}$$

$$\mathbf{a} = (3), \quad \mathbf{b} = (1 \quad 2), \quad \mathbf{d} = \begin{pmatrix} 2 \\ 1 \end{pmatrix}, \quad \mathbf{B} = \begin{pmatrix} 5 & 4 \\ 1 & 2 \end{pmatrix}$$

One useful partition is to write the matrix as either a row vector of column vectors or a column vector of row vectors

$$\mathbf{C} = \begin{pmatrix} 3 & 1 & 2 \\ 2 & 5 & 4 \\ 1 & 1 & 2 \end{pmatrix} = \begin{pmatrix} \mathbf{r}_1 \\ \mathbf{r}_2 \\ \mathbf{r}_3 \end{pmatrix}$$

A column vector whose elements are row vectors

$$\mathbf{r}_1 = (3 \quad 1 \quad 2)$$
$$\mathbf{r}_2 = (2 \quad 5 \quad 4)$$
$$\mathbf{r}_3 = (1 \quad 1 \quad 2)$$

$$\mathbf{C} = \begin{pmatrix} 3 & 1 & 2 \\ 2 & 5 & 4 \\ 1 & 1 & 2 \end{pmatrix} = (\mathbf{c}_1 \quad \mathbf{c}_2 \quad \mathbf{c}_3)$$

A row vector whose elements are column vectors

$$\mathbf{c}_1 = \begin{pmatrix} 3 \\ 2 \\ 1 \end{pmatrix}, \quad \mathbf{c}_2 = \begin{pmatrix} 1 \\ 5 \\ 1 \end{pmatrix}, \quad \mathbf{c}_3 = \begin{pmatrix} 2 \\ 4 \\ 2 \end{pmatrix}$$

Towards Matrix Multiplication: dot products

The dot (or inner) product of two vectors (both of length n) is defined as follows:

$$\mathbf{a} \cdot \mathbf{b} = \sum_{i=1}^{n} a_i b_i$$

Example:

$$\mathbf{a} = \begin{pmatrix} 1 \\ 2 \\ 3 \\ 4 \end{pmatrix} \quad \text{and} \quad \mathbf{b} = (4 \quad 5 \quad 7 \quad 9)$$

a ·b = 1*4 + 2*5 + 3*7 + 4*9 = 60

Matrices are compact ways to write systems of equations

$$
\begin{aligned}
5x_1 + 6x_2 + 4x_3 &= 6 \\
7x_1 - 3x_2 + 5x_3 &= -9 \\
-x_1 - x_2 + 6x_3 &= 12
\end{aligned}
$$

$$
\begin{pmatrix} 5 & 6 & 4 \\ 7 & -3 & 5 \\ -1 & -1 & 6 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 6 \\ -9 \\ 12 \end{pmatrix}
$$

$$
\mathbf{Ax} = \mathbf{c}, \qquad \text{or} \quad \mathbf{x} = \mathbf{A}^{-1}\mathbf{c}
$$

The least-squares solution for the linear model

$$y = \mu + \beta_1 z_1 + \cdots \beta_n z_n$$

yields the following system of equations for the β_i

$$
\begin{aligned}
\sigma(y, z_1) &= \beta_1 \sigma^2(z_1) + \beta_2 \sigma(z_1, z_2) + \cdots + \beta_n \sigma(z_1, z_n) \\
\sigma(y, z_2) &= \beta_1 \sigma(z_1, z_2) + \beta_2 \sigma^2(z_2) + \cdots + \beta_n \sigma(z_2, z_n) \\
&\vdots \\
\sigma(y, z_n) &= \beta_1 \sigma(z_1, z_n) + \beta_2 \sigma(z_2, z_n) + \cdots + \beta_n \sigma^2(z_n)
\end{aligned}
$$

This can be more compactly written in matrix form as

$$
\underbrace{\begin{pmatrix} \sigma^2(z_1) & \sigma(z_1, z_2) & \dots & \sigma(z_1, z_n) \\ \sigma(z_1, z_2) & \sigma^2(z_2) & \dots & \sigma(z_2, z_n) \\ \vdots & \vdots & \ddots & \vdots \\ \sigma(z_1, z_n) & \sigma(z_2, z_n) & \dots & \sigma^2(z_n) \end{pmatrix}}_{X^T X} \underbrace{\begin{pmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_n \end{pmatrix}}_{\beta} = \underbrace{\begin{pmatrix} \sigma(y, z_1) \\ \sigma(y, z_2) \\ \vdots \\ \sigma(y, z_n) \end{pmatrix}}_{X^T y}
$$

or, $\beta = (X^T X)^{-1} X^T y$

Matrix Multiplication:

The order in which matrices are multiplied affects the matrix product, e.g. $AB \neq BA$

For the product of two matrices to exist, the matrices must conform. For AB, the number of columns of A must equal the number of rows of B.

The matrix C = AB has the same number of rows as A and the same number of columns as B.

$$C_{(r\times c)} = A_{(r\times k)}\ B_{(k\times c)}$$

ij-th element of C is given by

$$C_{ij} = \sum_{l=1}^{k} A_{il} B_{lj}$$

Elements in the jth column of B

Elements in the ith row of matrix A

Outer indices given dimensions of resulting matrix, with r rows (A) and c columns (B)

$$C_{(rxc)} = A_{(rxk)}\ \ B_{(kxc)}$$

Inner indices must match columns of A = rows of B

Example: Is the product ABCD defined? If so, what is its dimensionality? Suppose

$$A_{3x5}\, B_{5x9}\, C_{9x6}\, D_{6x23}$$

Yes, defined, as inner indices match. Result is a 3 x 23 matrix (3 rows, 23 columns)

More formally, consider the product L = MN

Express the matrix M as a column vector of row vectors

$$\mathbf{M} = \begin{pmatrix} \mathbf{m_1} \\ \mathbf{m_2} \\ \vdots \\ \mathbf{m_r} \end{pmatrix} \quad \text{where} \quad \mathbf{m_i} = \begin{pmatrix} M_{i1} & M_{i2} & \cdots & M_{ic} \end{pmatrix}$$

Likewise express N as a row vector of column vectors

$$\mathbf{N} = \begin{pmatrix} \mathbf{n_1} & \mathbf{n_2} & \cdots & \mathbf{n_b} \end{pmatrix} \quad \text{where} \quad \mathbf{n_j} = \begin{pmatrix} N_{1j} \\ N_{2j} \\ \vdots \\ N_{cj} \end{pmatrix}$$

The ij-th element of L is the inner product of M's row i with N's column j

$$\mathbf{L} = \begin{pmatrix} \mathbf{m_1} \cdot \mathbf{n_1} & \mathbf{m_1} \cdot \mathbf{n_2} & \cdots & \mathbf{m_1} \cdot \mathbf{n_b} \\ \mathbf{m_2} \cdot \mathbf{n_1} & \mathbf{m_2} \cdot \mathbf{n_2} & \cdots & \mathbf{m_2} \cdot \mathbf{n_b} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{m_r} \cdot \mathbf{n_1} & \mathbf{m_r} \cdot \mathbf{n_2} & \cdots & \mathbf{m_r} \cdot \mathbf{n_b} \end{pmatrix}$$

Example

$$\mathbf{AB} = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} e & f \\ g & h \end{pmatrix} = \begin{pmatrix} ae+bg & af+bh \\ ce+dg & cf+dh \end{pmatrix}$$

Likewise

$$\mathbf{BA} = \begin{pmatrix} ae+cf & eb+df \\ ga+ch & gd+dh \end{pmatrix}$$

ORDER of multiplication matters! Indeed, consider $C_{3x5}\,D_{5x5}$ which gives a 3 x 5 matrix, versus $D_{5x5}\,C_{3x5}$, which is not defined.

Matrix multiplication in R

```
> A<-matrix(c(1,2,3,4),nrow=2)
> B<-matrix(c(4,5,6,7),nrow=2)
> A
     [,1] [,2]
[1,]    1    3
[2,]    2    4
> B
     [,1] [,2]
[1,]    4    6
[2,]    5    7
> A %*% B
     [,1] [,2]
[1,]   19   27
[2,]   28   40
```

R fills in the matrix from the list c by filling in as columns, here with 2 rows (nrow=2)

Entering A or B displays what was entered (always a good thing to check)

The command %*% is the R code for the multiplication of two matrices

On your own: What is the matrix resulting from BA? What is A if nrow=1 or nrow=4 is used?

The Transpose of a Matrix

The transpose of a matrix exchanges the rows and columns, $A^T_{ij} = A_{ji}$

Useful identities

$(AB)^T = B^T A^T$

$(ABC)^T = C^T B^T A^T$

$$\mathbf{a} = \begin{pmatrix} a_1 \\ \vdots \\ a_n \end{pmatrix} \qquad \mathbf{b} = \begin{pmatrix} b_1 \\ \vdots \\ b_n \end{pmatrix}$$

Inner product = $a^T b = a^T_{(1 \times n)} b_{(n \times 1)}$

Indices match, matrices conform

Dimension of resulting product is 1 X 1 (i.e. a scalar)

$$\begin{pmatrix} a_1 & \cdots & a_n \end{pmatrix} \begin{pmatrix} b_1 \\ \vdots \\ b_n \end{pmatrix} = \mathbf{a}^T\mathbf{b} = \sum_{i=1}^{n} a_i b_i$$

Note that $b^T a = (b^T a)^T = a^T b$

Outer product = ab^T = $a_{(n \times 1)}\, b^T_{(1 \times n)}$

Resulting product is an n x n matrix

$$\begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{pmatrix} \begin{pmatrix} b_1 & b_2 & \cdots & b_n \end{pmatrix}$$

$$= \begin{pmatrix} a_1 b_1 & a_1 b_2 & \dots & a_1 b_n \\ a_2 b_1 & a_2 b_2 & \dots & a_2 b_n \\ \vdots & \vdots & \ddots & \vdots \\ a_n b_1 & a_n b_2 & \dots & a_n b_{bn} \end{pmatrix}$$

Multidimensional Taylor series

Suppose we let $f(\mathbf{x})$ be a scalar (single-dimension) function of a column vector, $\mathbf{x} = (x_1, \cdots, x_n)^T$, of n variables. The **gradient** (or **gradient vector**) of f with respect to $\mathbf{x}$ is obtained by taking partial derivatives of the function with respect to each variable. In matrix notation, the gradient operator is denoted by

$$\nabla_{\mathbf{x}}[f] = \frac{\partial f}{\partial \mathbf{x}} = \begin{pmatrix} \frac{\partial f}{\partial x_1} \\ \vdots \\ \frac{\partial f}{\partial x_n} \end{pmatrix}$$

The gradient at a point, $\mathbf{x}_o$, corresponds to a vector indicating the direction of local steepest ascent of the function at that point (the multivariate slope of f at $\mathbf{x}_o$).

In univariate calculus, the local extrema of a function occur when its slope (first derivative) is zero. The multivariate extension is that the gradient vector is zero, so the slope of the function with respect to all variables is zero. A point $\mathbf{x}_e$ where this occurs is called a **stationary** or **equilibrium** point, and corresponds to either a local maximum, minimum, saddle point, or inflection point. As with the calculus of single variables, determining which of these cases is correct depends on the second derivative. With n variables, the appropriate generalization is the **Hessian** matrix

$$\mathbf{H}_{\mathbf{x}}[f] = \nabla_{\mathbf{x}}\left[\left(\nabla_{\mathbf{x}}[f]\right)^T\right] = \frac{\partial^2 f}{\partial \mathbf{x}\,\partial \mathbf{x}^T} = \begin{pmatrix} \dfrac{\partial^2 f}{\partial x_1^2} & \cdots & \dfrac{\partial^2 f}{\partial x_1\,\partial x_n} \\ \vdots & \ddots & \vdots \\ \dfrac{\partial^2 f}{\partial x_1\,\partial x_n} & \cdots & \dfrac{\partial^2 f}{\partial x_n^2} \end{pmatrix} \tag{A6.5}$$

Note that this is the **outer product** of $\nabla_{\mathbf{x}}[f]$ with itself. Recall for an n-dimensional column vector $\mathbf{a}_{n\times 1}$ that while the inner product, $\mathbf{a}_{1\times n}^T\mathbf{a}_{n\times 1} = \sum a_i$, returns a 1×1 matrix (a scalar), the outer product, $\mathbf{a}_{n\times 1}\mathbf{a}_{1\times n}^T$, returns an $n \times n$ matrix whose ijth element is $a_i a_j$, or (in our case)

$$\mathbf{H}_{ij} = \frac{\partial\left(f(\mathbf{x})/\partial x_i\right)}{\partial x_j} = \frac{\partial^2 f(\mathbf{x})}{\partial x_i\,\partial x_j}$$

To see how the Hessian matrix determines the nature of equilibrium points, a slight digression on the **multidimensional Taylor series** is needed. Consider the (second-order) Taylor series of a scalar function of n variables, $f(x_1, \cdots, x_n)$, expanded about the point $\mathbf{x}_o$,

$$f(\mathbf{x}) \simeq f(\mathbf{x}_o) + \sum_{i=1}^{n}(x_i - x_{o,i})\frac{\partial f}{\partial x_i} + \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}(x_i - x_{o,i})(x_j - x_{o,j})\frac{\partial^2 f}{\partial x_i\,\partial x_j} + \cdots \tag{A6.7a}$$

where all partials are evaluated at $\mathbf{x}_o$. If we note that the first sum is the inner product of the gradient and $(\mathbf{x} - \mathbf{x}_o)$, and the double sum is a quadratic product involving the Hessian, we can express Equation A6.7a in matrix form as

$$f(\mathbf{x}) \simeq f(\mathbf{x}_o) + \nabla^T(\mathbf{x} - \mathbf{x}_o) + \frac{1}{2}(\mathbf{x} - \mathbf{x}_o)^T\mathbf{H}\,(\mathbf{x} - \mathbf{x}_o) \tag{A6.7b}$$

where ∇ and $\mathbf{H}$ are the gradient and Hessian of f with respect to $\mathbf{x}$ when evaluated at $\mathbf{x}_o$,

$$\nabla \equiv \nabla_{\mathbf{x}}[f]\big|_{\mathbf{x}=\mathbf{x}_o} \qquad \text{and} \qquad \mathbf{H} \equiv \mathbf{H}_{\mathbf{x}}[f]\big|_{\mathbf{x}=\mathbf{x}_o}$$

R code for transposition

```
> t(A)
     [,1] [,2]
[1,]    1    2
[2,]    3    4
```

t(A) = transpose of A

```
> a<-matrix(c(1,2,3),nrow=3)
> a
     [,1]
[1,]    1
[2,]    2
[3,]    3
> t(a) %*% a
     [,1]
[1,]   14
> a %*% t(a)
     [,1] [,2] [,3]
[1,]    1    2    3
[2,]    2    4    6
[3,]    3    6    9
```

Enter the column vector a

Compute inner product $\mathbf{a}^T\mathbf{a}$

Compute outer product $\mathbf{a}\mathbf{a}^T$

Solving equations

- The identity matrix **I**
  - Serves the same role as 1 in scalar algebra, e.g., a*1=1*a =a, with **AI=IA= A**
- The inverse matrix $\mathbf{A}^{-1}$ (IF it exists)
  - Defined by $\mathbf{A}\,\mathbf{A}^{-1} = \mathbf{I}$, $\mathbf{A}^{-1}\mathbf{A} = \mathbf{I}$
  - Serves the same role as scalar division
    - To solve ax = c, multiply both sides by (1/a) to give:
    - (1/a)*ax = (1/a)c or (1/a)*a*x = 1*x = x,
    - Hence x = (1/a)c
    - To solve $\mathbf{Ax} = \mathbf{c}$, $A^{-1}\mathbf{Ax} = A^{-1}\,\mathbf{c}$
    - Or $\mathbf{A}^{-1}\mathbf{Ax} = \mathbf{Ix} = \mathbf{x} = A^{-1}\,\mathbf{c}$

The Identity Matrix, I

The identity matrix serves the role of the number 1 in matrix multiplication: AI =A, IA = A

I is a square diagonal matrix, with all diagonal elements being one, all off-diagonal elements zero.

$$I_{ij} = \begin{cases} 1 \text{ for } i = j \\ 0 \text{ otherwise} \end{cases}$$

$$\mathbf{I}_{3x3} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

The Identity Matrix in R

diag(k), where k is an integer, return the k x k I matix

```
> I<-diag(4)
> I
     [,1] [,2] [,3] [,4]
[1,]    1    0    0    0
[2,]    0    1    0    0
[3,]    0    0    1    0
[4,]    0    0    0    1
> I2 <-diag(2)
> I2
     [,1] [,2]
[1,]    1    0
[2,]    0    1
```

The Inverse Matrix, A^{-1}

For a square matrix A, define its Inverse A^{-1}, as the matrix satisfying

$$\mathbf{A}^{-1}\mathbf{A} = \mathbf{A}\mathbf{A}^{-1} = \mathbf{I}$$

For $\mathbf{A} = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$ $\mathbf{A}^{-1} = \frac{1}{ad - bc}\begin{pmatrix} d & -b \\ -c & a \end{pmatrix}$

If this quantity (the determinant) is zero, the inverse does not exist.

If det(A) is not zero, A^{-1} exists and A is said to be non-singular. If det(A) = 0, A is singular, and no *unique* inverse exists (generalized inverses do)

Generalized inverses, and their uses in solving systems of equations, are discussed in Appendix 3 of Lynch & Walsh

A^{-} is the typical notation to denote the G-inverse of a matrix

When a G-inverse is used, *provided* the system is consistent, then some of the variables have a family of solutions (e.g., $x_1 = 2$, but $x_2 + x_3 = 6$)

Inversion in R

solve(A) computes A^{-1}

det(A) computes determinant of A

```
> A
     [,1] [,2]
[1,]    1    3
[2,]    2    4
> solve(A)
     [,1] [,2]
[1,]   -2  1.5
[2,]    1 -0.5
> solve(A) %*% A
     [,1]          [,2]
[1,]    1 -8.881784e-16
[2,]    0  1.000000e+00
> det(A)
[1] -2
```

Using A entered earlier

Compute A^{-1}

Showing that $A^{-1} A = I$

Computing determinant of A

Example 8.5. To see further connections between the determinant and the solution to a set of equations, consider the following two systems of equations:

$$\text{Set one:} \quad \begin{aligned} x_1 + x_2 &= 1 \\ 2x_1 + 2x_2 &= 2 \end{aligned} \qquad \text{Set two:} \quad \begin{aligned} 0.9999 \cdot x_1 + x_2 &= 1 \\ 2x_1 + 2x_2 &= 2 \end{aligned}$$

The determinant for the coefficient matrix associated with set one is zero, and there is no unique solution, rather a line of solutions, $x_1 = 1 - x_2$. In contrast, the determinant for the matrix associated with set two is nonzero, hence its inverse exists and there is a unique solution. However, the determinant nearly zero, 0.0002. Such a matrix is said to be **nearly singular**, meaning that although the two sets of equations are distinct, they overlap so closely that there is little additional information from one (or more) of the equations. For this set of equations,

$$\mathbf{A}^{-1} = \begin{pmatrix} -10,000 & 5000 \\ -10,000 & -4999.5 \end{pmatrix}, \quad \mathbf{x} = \begin{pmatrix} -3.63 \times 10^{-12} \\ 1 \end{pmatrix} \simeq \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

While there *technically* is a unique solution, it is *extremely* sensitive to the coefficients in the set of equations, and a very small change (such as through measurement error) can dramatically change the solution. For example, replacing the first equation by $x_1 + 0.9999 \cdot x_2 = 1$, yields the solution of $x_1 = 1, x_2 \simeq 0$.

Useful identities

$$(A^T)^{-1} = (A^{-1})^T$$

$$(AB)^{-1} = B^{-1} A^{-1}$$

For a diagonal matrix **D**, then det (**D**), which is also denoted by |**D**|, = product of the diagonal elements

Also, the determinant of any square matrix **A**, det(**A**), is simply the product of the eigenvalues λ of **A**, which statisfy

$$\mathbf{Ae} = \lambda \mathbf{e}$$

If A is n x n, solutions to λ are an n-degree polynomial. **e** is the eigenvector associated with λ. If any of the roots to the equation are zero, A^{-1} is not defined. In this case, for some linear combination **b**, we have **Ab** = 0.

Variance-Covariance matrix

- A very important square matrix is the variance-covariance matrix **V** associated with a vector **x** of random variables.
- $V_{ij} = \mathrm{Cov}(x_i,x_j)$, so that the i-th diagonal element of **V** is the variance of x_i, and off-diagonal elements are covariances
- **V** is a symmetric, square matrix

The trace

The trace, tr(**A**) or trace(**A**), of a square matrix A is simply the sum of its diagonal elements

The importance of the trace is that it equals the sum of the eigenvalues of **A**, tr(**A**) = $\sum \lambda_i$

For a covariance matrix **V**, tr(V) measures the total amount of variation in the variables

λ_i / tr(**V**) is the fraction of the total variation in x contained in the linear combination $\mathbf{e}_i^T\mathbf{x}$, where $\mathbf{e}_i$, the i-th principal component of **V** is also the i-th eigenvector of **V** ($\mathbf{V}\mathbf{e}_i = \lambda_i\, \mathbf{e}_i$)

Eigenstructure in R

`eigen(A)` returns the eigenvalues and vectors of A

```
> V<-matrix(c(10,-5,10,-5,20,0,10,0,30),nrow=3)
> V
     [,1] [,2] [,3]
[1,]   10   -5   10
[2,]   -5   20    0
[3,]   10    0   30
> eigen(V)
$values
[1] 34.410103 21.117310  4.472587

$vectors
           [,1]       [,2]       [,3]
[1,]  0.3996151  0.2117936  0.8918807
[2,] -0.1386580 -0.9477830  0.2871955
[3,]  0.9061356 -0.2384340 -0.3493816
```

PC 1

Trace = 60

PC 1 accounts for 34.4/60 = 57% of all the variation

$0.400^* x_1 - 0.139^*x_2 + 0.906^*x_3$

$$f(\mathbf{x}) \simeq f(\mathbf{x}_o) + \nabla^T (\mathbf{x} - \mathbf{x}_o) + \frac{1}{2}(\mathbf{x} - \mathbf{x}_o)^T \mathbf{H} (\mathbf{x} - \mathbf{x}_o) \tag{A6.7b}$$

where ∇ and $\mathbf{H}$ are the gradient and Hessian of f with respect to $\mathbf{x}$ when evaluated at $\mathbf{x}_o$,

$$\nabla \equiv \nabla_{\mathbf{x}}[\,f\,]\,\big|_{\mathbf{x}=\mathbf{x}_o} \qquad \text{and} \qquad \mathbf{H} \equiv \mathbf{H}_{\mathbf{x}}[\,f\,]\,\big|_{\mathbf{x}=\mathbf{x}_o}$$

At an equilibrium point, $\widehat{\mathbf{x}}$, all first partials are zero, so $(\nabla_{\mathbf{x}}[\,f\,])^T$ is evaluated at $\widehat{\mathbf{x}}$ is a vector of length zero. Whether this point is a maximum or minimum is then determined by the quadratic product involving the Hessian when evaluated at $\widehat{\mathbf{x}}$. Consider a vector, $\mathbf{d}$, of a small change from the equilibrium point

$$f(\widehat{\mathbf{x}} + \mathbf{d}) - f(\widehat{\mathbf{x}}) \simeq \frac{1}{2} \cdot \mathbf{d}^T\, \mathbf{H}\, \mathbf{d} \tag{A6.8a}$$

Because $\mathbf{H}$ is a symmetric matrix, we can diagonalize it and apply a canonical transformation (Equation A5.17a) to simplify the quadratic product in Equation A6.8a, which returns

$$f(\widehat{\mathbf{x}} + \mathbf{d}) - f(\widehat{\mathbf{x}}) \simeq \frac{1}{2} \sum_{i=1}^{n} \lambda_i y_i^2 \tag{A6.8b}$$

Stable if all $|\lambda_i| < 1$

Leslie matrix: *Age-structured growth*

$$\mathbf{L} = \begin{pmatrix} b_1 & b_2 & \cdots & b_{k-1} & b_k \\ \ell_1 & 0 & \cdots & 0 & 0 \\ 0 & \ell_2 & \cdots & 0 & 0 \\ \vdots & & \ddots & 0 & 0 \\ 0 & 0 & \cdots & \ell_{k-1} & 0 \end{pmatrix} \tag{29.3a}$$

If $\mathbf{n}(t)$ is a vector of the number of individuals in each age/stage class at time t, then $\mathbf{n}(t+1) = \mathbf{L}\mathbf{n}(t)$. The **asymptotic growth rate**, λ, for this population is the largest eigenvalue of $\mathbf{L}$, while its associated eigenvector is the stable age distribution. The Leslie matrix is just one type of life-history projection matrix. More generally, the life history of a species may be more accurately defined by **stages**, rather than *ages*. For example, a perennial plant could spend many years in a rosette stage before flowering. Life-history graphs are a more general approach for categorizing such stage-structured organisms. In an age-structured model, individuals increase in age at each step, but in a stage-structured model, an individual can *remain* in the same stage on the next step (e.g., stays as a rosette), generating a loop in the graph (an arrow that circles back to itself), and hence a nonzero diagonal element, L_{ii}, representing the chance of remaining in stage i in the next step (Caswell 1989, 2001).

$$\mathbf{L}_3 = \begin{pmatrix} 0 & 10 & 10 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}$$

```
> L<-matrix(c(0,1,0,10,0,1,10,0,0),nrow=3)
> L
     [,1] [,2] [,3]
[1,]    0   10   10
[2,]    1    0    0
[3,]    0    1    0
> eigen(L)
eigen() decomposition
$values
[1]  3.577089 -2.423622 -1.153467

$vectors
           [,1]       [,2]       [,3]
[1,] 0.96035836 -0.9131654  0.6570260
[2,] 0.26847480  0.3767771 -0.5696095
[3,] 0.07505398 -0.1554603  0.4938237
```

Growth rate = 3.58
1 class % = 73.7
= .96/(.96+.27+0.8)
2 class % = 20.6
3 class % = 5.8

Quadratic and Bilinear Forms

Quadratic product: for $A_{n \times n}$ and $x_{n \times 1}$

$$\mathbf{x}^T\mathbf{A}\mathbf{x} = \sum_{i=1}^{n}\sum_{j=1}^{n} a_{ij}x_ix_j \quad \text{Scalar } (1 \times 1)$$

Bilinear Form (generalization of quadratic product)
for $A_{m \times n}$, $a_{n \times 1}$, $b_{m \times 1}$ their bilinear form is $b^T_{1 \times m} A_{m \times n} a_{n \times 1}$

$$\mathbf{b}^T\mathbf{A}\mathbf{a} = \sum_{i=1}^{m}\sum_{j=1}^{n} A_{ij}b_ia_j$$

Note that $b^TA\,a = a^TA^T b$

Covariance Matrices for Transformed Variables

What is the variance of the linear combination,
$c_1x_1 + c_2x_2 + \ldots + c_nx_n$? (note this is a scalar)

$$\sigma^2\left(\mathbf{c}^T\mathbf{x}\right) = \sigma^2\left(\sum_{i=1}^{n} c_i x_i\right) = \sigma\left(\sum_{i=1}^{n} c_i x_i, \sum_{j=1}^{n} c_j x_j\right)$$
$$= \sum_{i=1}^{n}\sum_{j=1}^{n} \sigma\left(c_i x_i, c_j x_j\right) = \sum_{i=1}^{n}\sum_{j=1}^{n} c_i c_j \sigma\left(x_i, x_j\right)$$
$$= \mathbf{c}^T\mathbf{V}\mathbf{c}$$

Likewise, the covariance between two linear combinations can be expressed as a bilinear form,

$$\sigma(\mathbf{a}^T\mathbf{x}, \mathbf{b}^T\mathbf{x}) = \mathbf{a}^T\mathbf{V}\,\mathbf{b}$$

Example: Suppose the variances of x_1, x_2, and x_3 are 10, 20, and 30. x_1 and x_2 have a covariance of -5, x_1 and x_3 of 10, while x_2 and x_3 are uncorrelated.

What are the variances of the indices
$y_1 = x_1 - 2x_2 + 5x_3$ and $y_2 = 6x_2 - 4x_3$?

$$\mathbf{V} = \begin{pmatrix} 10 & -5 & 10 \\ -5 & 20 & 0 \\ 10 & 0 & 30 \end{pmatrix}, \quad \mathbf{c}_1 = \begin{pmatrix} 1 \\ -2 \\ 5 \end{pmatrix}, \quad \mathbf{c}_2 = \begin{pmatrix} 0 \\ 6 \\ -4 \end{pmatrix}$$

$\mathrm{Var}(y_1) = \mathrm{Var}(c_1^T x) = c_1^T\,\mathrm{Var}(x)\,c_1 = 960$

$\mathrm{Var}(y_2) = \mathrm{Var}(c_2^T x) = c_2^T\,\mathrm{Var}(x)\,c_2 = 1200$

$\mathrm{Cov}(y_1, y_2) = \mathrm{Cov}(c_1^T x, c_2^T x) = c_1^T\,\mathrm{Var}(x)\,c_2 = -910$

Homework: use R to compute the above values

The Multivariate Normal Distribution (MVN)

Consider the pdf for n independent normal random variables, the ith of which has mean μ_i and variance σ^2_i

$$p(\mathbf{x}) = \prod_{i=1}^{n} (2\pi)^{-1/2} \sigma_i^{-1} \exp\left(- \frac{(x_i - \mu_i)^2}{2\,\sigma_i^2} \right)$$
$$= (2\pi)^{-n/2} \left(\prod_{i=1}^{n} \sigma_i \right)^{-1} \exp\left(- \sum_{i=1}^{n} \frac{(x_i - \mu_i)^2}{2\,\sigma_i^2} \right)$$

This can be expressed more compactly in matrix form

Define the covariance matrix V for the vector x of the n normal random variable by

$$\mathbf{V} = \begin{pmatrix} \sigma_1^2 & 0 & \cdots & 0 \\ 0 & \sigma_2^2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & \cdots & \sigma_n^2 \end{pmatrix} \qquad |\mathbf{V}| = \prod_{i=1}^{n} \sigma_i^2$$

Define the mean vector μ by gives

$$\boldsymbol{\mu} = \begin{pmatrix} \mu_1 \\ \mu_2 \\ \vdots \\ \mu_n \end{pmatrix}$$

$$\sum_{i=1}^{n} \frac{(x_i - \mu_i)^2}{\sigma_i^2} = (\mathbf{x} - \boldsymbol{\mu})^T \mathbf{V}^{-1} (\mathbf{x} - \boldsymbol{\mu})$$

Hence in matrix from the MVN pdf becomes

$$p(\mathbf{x}) = (2\pi)^{-n/2} |\mathbf{V}|^{-1/2} \exp\left[-\frac{1}{2} (\mathbf{x} - \boldsymbol{\mu})^T \mathbf{V}^{-1} (\mathbf{x} - \boldsymbol{\mu}) \right]$$

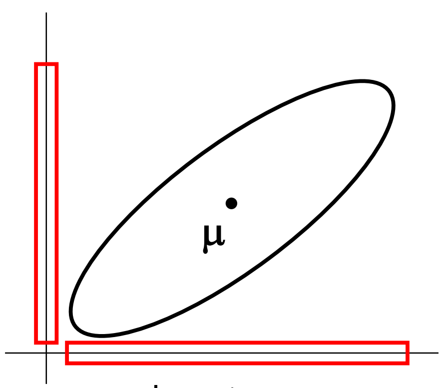
Notice this holds for any vector μ and symmetric positive-definite matrix V, as | V | > 0.

The multivariate normal

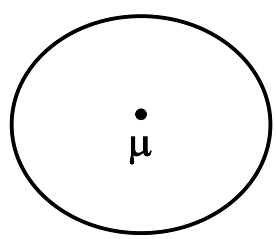
- Just as a univariate normal is defined by its mean and spread, a multivariate normal is defined by its mean vector $\boldsymbol{\mu}$ (also called the centroid) and variance-covariance matrix $\mathbf{V}$

Vector of means $\boldsymbol{\mu}$ determines location

Spread (geometry) about $\boldsymbol{\mu}$ determined by $\mathbf{V}$



x_1, x_2 equal variances, positively correlated

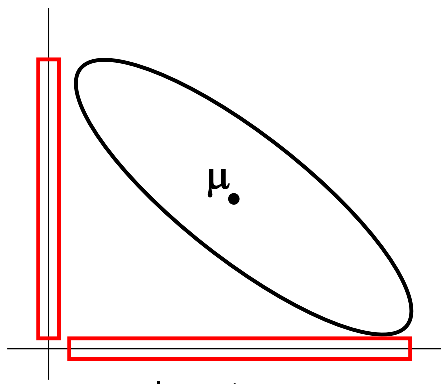


x_1, x_2 equal variances, uncorrelated

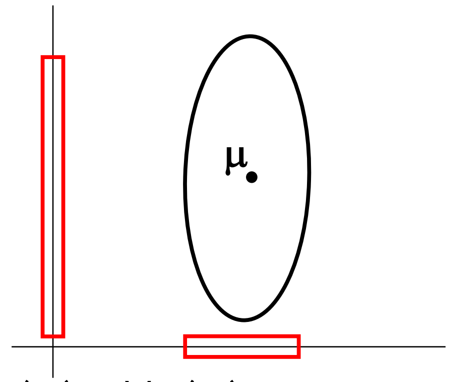
Eigenstructure (the eigenvectors and their corresponding eigenvalues) determines the geometry of $\mathbf{V}$.

Vector of means $\boldsymbol{\mu}$ determines location

Spread (geometry) about $\boldsymbol{\mu}$ determined by **V**



x_1, x_2 equal variances, negatively correlated

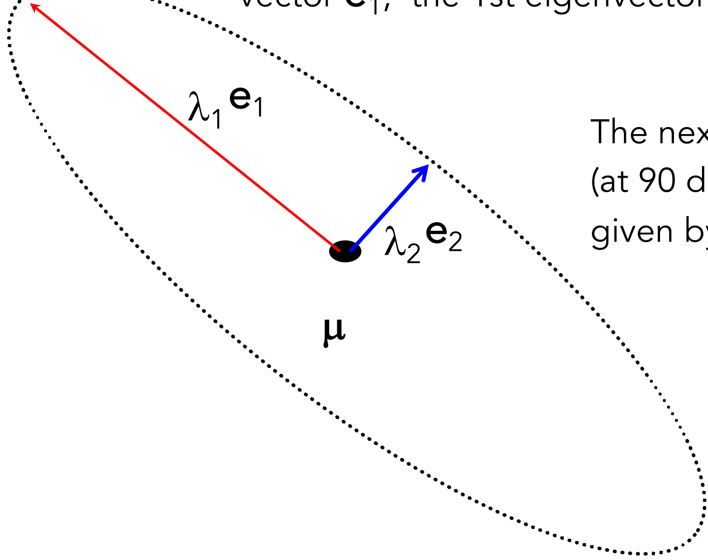


$Var(x_1) < Var(x_2)$, uncorrelated

Positive tilt = positive correlations
Negative tilt = negative correlation
No tilt = uncorrelated

Eigenstructure of V

The direction of the largest axis of variation is given by the unit-length vector $\mathbf{e}_1$, the 1st eigenvector of V.



The next largest axis of orthogonal (at 90 degrees from) $\mathbf{e}_1$, is given by $\mathbf{e}_2$, the 2nd eigenvector

Principal components

- The principal components (or PCs) of a covariance matrix define the axes of variation.
  - PC1 is the direction (linear combination $\mathbf{c}^T\mathbf{x}$) that explains the most variation.
  - PC2 is the next largest direction (at 90degree from PC1), and so on
- PC_i = ith eigenvector of V
- Fraction of variation accounted for by PCi = λ_i / trace(V)
- If V has a few large eigenvalues, most of the variation is distributed along a few linear combinations (axis of variation)
- The singular value decomposition is the generalization of this idea to nonsquare matrices

Properties of the MVN - I

1) If **x** is MVN, any subset of the variables in **x** is also MVN

2) If **x** is MVN, any linear combination of the elements of **x** is also MVN. If **x** ~ MVN(μ,V)

$$\text{for} \quad \mathbf{y} = \mathbf{x} + \mathbf{a}, \qquad \mathbf{y} \text{ is } \mathrm{MVN}_n(\boldsymbol{\mu} + \mathbf{a}, \mathbf{V})$$

$$\text{for} \quad y = \mathbf{a}^T\mathbf{x} = \sum_{k=1}^{n} a_i x_i, \qquad y \text{ is } \mathrm{N}(\mathbf{a}^T\boldsymbol{\mu}, \mathbf{a}^T\mathbf{V}\,\mathbf{a})$$

$$\text{for} \quad \mathbf{y} = \mathbf{A}\mathbf{x}, \qquad \mathbf{y} \text{ is } \mathrm{MVN}_m\left(\mathbf{A}\boldsymbol{\mu}, \mathbf{A}^T\mathbf{V}\mathbf{A}\right)$$

Properties of the MVN - II

3) Conditional distributions are also MVN. Partition x into two components, x_1 (m dimensional column vector) and x_2 (n-m dimensional column vector)

$$\mathbf{x} = \begin{pmatrix} \mathbf{x_1} \\ \mathbf{x_2} \end{pmatrix} \quad \boldsymbol{\mu} = \begin{pmatrix} \boldsymbol{\mu_1} \\ \boldsymbol{\mu_2} \end{pmatrix} \quad \text{and} \quad \mathbf{V} = \begin{pmatrix} \mathbf{V_{x_1x_1}} & \mathbf{V_{x_1x_2}} \\ \mathbf{V^T_{x_1x_2}} & \mathbf{V_{x_2x_2}} \end{pmatrix}$$

x_1 | x_2 is MVN with m-dimensional mean vector

$$\boldsymbol{\mu}_{\mathbf{x_1|x_2}} = \boldsymbol{\mu_1} + \mathbf{V_{x_1x_2}}\mathbf{V^{-1}_{x_2x_2}}(\mathbf{x_2} - \boldsymbol{\mu_2})$$

and m x m covariance matrix

$$\mathbf{V_{x_1|x_2}} = \mathbf{V_{x_1x_1}} - \mathbf{V_{x_1x_2}}\mathbf{V^{-1}_{x_2x_2}}\mathbf{V^{T}_{x_1x_2}}$$

Properties of the MVN - III

4) If x is MVN, the regression of any subset of x on another subset is linear and homoscedastic

$$\mathbf{x_1} = \boldsymbol{\mu}_{\mathbf{x_1|x_2}} + \mathbf{e}$$

$$= \boldsymbol{\mu_1} + \mathbf{V_{x_1x_2}}\mathbf{V^{-1}_{x_2x_2}}(\mathbf{x_2} - \boldsymbol{\mu_2}) + \mathbf{e}$$

Where e is MVN with mean vector **0** and variance-covariance matrix $\mathbf{V_{x_1|x_2}}$

$$\boldsymbol{\mu}_1 + \mathbf{V}_{\mathbf{x}_1\mathbf{x}_2}\mathbf{V}_{\mathbf{x}_2\mathbf{x}_2}^{-1}(\mathbf{x}_2 - \boldsymbol{\mu}_2) + \mathbf{e}$$

The regression is linear because it is a linear function of x_2

The regression is homoscedastic because the variance-covariance matrix for e does not depend on the value of the x's

$$\mathbf{V}_{\mathbf{x}_1|\mathbf{x}_2} = \mathbf{V}_{\mathbf{x}_1\mathbf{x}_1} - \mathbf{V}_{\mathbf{x}_1\mathbf{x}_2}\mathbf{V}_{\mathbf{x}_2\mathbf{x}_2}^{-1}\mathbf{V}_{\mathbf{x}_1\mathbf{x}_2}^{T}$$

All these matrices are constant, and hence the same for any value of x

Example: Regression of Offspring value on Parental values

Assume the vector of offspring value and the values of both its parents is MVN. Then from the correlations among (outbred) relatives,

$$\begin{pmatrix} z_o \\ z_s \\ z_d \end{pmatrix} \sim \text{MVN}\left[\begin{pmatrix} \mu_o \\ \mu_s \\ \mu_d \end{pmatrix}, \sigma_z^2 \begin{pmatrix} 1 & h^2/2 & h^2/2 \\ h^2/2 & 1 & 0 \\ h^2/2 & 0 & 1 \end{pmatrix} \right]$$

Let $\mathbf{x_1} = (\, z_o \,), \quad \mathbf{x_2} = \begin{pmatrix} z_s \\ z_d \end{pmatrix}$

$$\mathbf{V}_{\mathbf{x}_1,\mathbf{x}_1} = \sigma_z^2, \quad \mathbf{V}_{\mathbf{x}_1,\mathbf{x}_2} = \frac{h^2\sigma_z^2}{2}(\,1 \quad 1\,), \quad \mathbf{V}_{\mathbf{x}_2,\mathbf{x}_2} = \sigma_z^2\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

$$= \boldsymbol{\mu}_1 + \mathbf{V}_{\mathbf{x}_1\mathbf{x}_2}\mathbf{V}_{\mathbf{x}_2\mathbf{x}_2}^{-1}(\mathbf{x}_2 - \boldsymbol{\mu}_2) + \mathbf{e}$$

Regression of Offspring value on Parental values (cont.)

$$= \boldsymbol{\mu_1} + \mathbf{V_{x_1x_2}} \mathbf{V_{x_2x_2}^{-1}} (\mathbf{x_2} - \boldsymbol{\mu_2}) + \mathbf{e}$$

$$\mathbf{V_{x_1,x_1}} = \sigma_z^2, \quad \mathbf{V_{x_1,x_2}} = \frac{h^2\sigma_z^2}{2} \begin{pmatrix} 1 & 1 \end{pmatrix}, \quad \mathbf{V_{x_2,x_2}} = \sigma_z^2 \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

Hence,

$$\begin{aligned} z_o &= \mu_o + \frac{h^2\sigma_z^2}{2} \begin{pmatrix} 1 & 1 \end{pmatrix} \sigma_z^{-2} \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} z_s - \mu_s \\ z_d - \mu_d \end{pmatrix} + e \\ &= \mu_o + \frac{h^2}{2} (z_s - \mu_s) + \frac{h^2}{2} (z_d - \mu_d) + e \end{aligned}$$

Where e is normal with mean zero and variance

$$\mathbf{V_{x_1|x_2}} = \mathbf{V_{x_1x_1}} - \mathbf{V_{x_1x_2}} \mathbf{V_{x_2x_2}^{-1}} \mathbf{V_{x_1x_2}^{T}}$$

$$\begin{aligned} \sigma_e^2 &= \sigma_z^2 - \frac{h^2\sigma_z^2}{2} \begin{pmatrix} 1 & 1 \end{pmatrix} \sigma_z^{-2} \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \frac{h^2\sigma_z^2}{2} \begin{pmatrix} 1 \\ 1 \end{pmatrix} \\ &= \sigma_z^2 \left(1 - \frac{h^4}{2}\right) \end{aligned}$$

Additional R matrix commands

Operator or Function	Description
A * B	Element-wise multiplication
A %*% B	Matrix multiplication
A %o% B	Outer product. AB'
crossprod(A,B) crossprod(A)	A'B and A'A respectively.
t(A)	Transpose
diag(x)	Creates diagonal matrix with elements of x in the principal diagonal
diag(A)	Returns a vector containing the elements of the principal diagonal
diag(k)	If k is a scalar, this creates a k x k identity matrix. Go figure.
solve(A, b)	Returns vector x in the equation b = Ax (i.e., $A^{-1}b$)
solve(A)	Inverse of A where A is a square matrix.
ginv(A)	Moore-Penrose Generalized Inverse of A. ginv(A) requires loading the MASS package.
y<-eigen(A)	y$val are the eigenvalues of A y$vec are the eigenvectors of A
y<-svd(A)	Single value decomposition of A. y$d = vector containing the singular values of A y$u = matrix with columns contain the left singular vectors of A y$v = matrix with columns contain the right singular vectors of A

Additional R matrix commands (cont)

R <- chol(A)	Choleski factorization of **A.** Returns the upper triangular factor, such that **R'R** = **A.**
y <- qr(A)	QR decomposition of **A.** **y$qr** has an upper triangle that contains the decomposition and a lower triangle that contains information on the Q decomposition. **y$rank** is the rank of A. **y$qraux** a vector which contains additional information on Q. **y$pivot** contains information on the pivoting strategy used.
cbind(A,B,...)	Combine matrices(vectors) horizontally. Returns a matrix.
rbind(A,B,...)	Combine matrices(vectors) vertically. Returns a matrix.
rowMeans(A)	Returns vector of row means.
rowSums(A)	Returns vector of row sums.
colMeans(A)	Returns vector of column means.
colSums(A)	Returns vector of coumn means.

Additional references

- Lynch & Walsh (1998)
  - Chapter 8 (intro to matrices)
  - Appendix 3 (G-inverses_
- Walsh and Lynch (2018)
  - Appendix 5 (Matrix geometry)
  - Appendix 6 (Matrix derivatives)