Linear Algebra
A quick wrap-up

Ott Toomet

April 4, 2017

Where We Stand

- Introduction
- Methods
  - python, pandas
  - Linear algebra
  - Probability and statistics
- Guest speaker
- Linear regression
  - Causality
- ML
  - Experiment design
  - Nearest neighbors
- Methods
  - Gradient Descent
  - Maximum Likelihood, logit
  - regularization
- ML
  - Naive bayes
  - PCA/dimensionality reduction
  - Clusters & recommenders
  - Trees and forests
  - Neural networks
- Wrap-up

What Is Linear Algebra

- Math with vectors and matrices
  - includes "addition" and "multiplication" $\Rightarrow$ it is an "algebra"
  - Multiplication and addition are *linear*:

$$A \times (B + C) = A \times B + A \times C \qquad \text{(distributive)}$$
$$A \times (\lambda \cdot C) = \lambda \cdot (A \times C)$$

$\times$ matrix multiplication
$\cdot$ *scalar multiplication*

Why Is It Useful

Express complex relationships in a compact way

- Example: equation system with 3 variables
- Non-matrix notation

$$\begin{aligned} x_1 + 2x_2 &= 7 \\ -x_2 + x_3 &= 6 \\ x_1 - 4x_2 + x_3 &= -2 \end{aligned}$$

- Matrix notation

$$\begin{pmatrix} 1 & 2 & 0 \\ 0 & -1 & 1 \\ 1 & -4 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 7 \\ 6 \\ -2 \end{pmatrix}$$

It Is Easier to Solve

- Non-Matrix: long and tedious and error-prone `http://www.sosmath.com/soe/SE311105/SE311105.html`
- Matrix:

$$\begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 1 & 2 & 0 \\ 0 & -1 & 1 \\ 1 & -4 & 1 \end{pmatrix}^{-1} \begin{pmatrix} 7 \\ 6 \\ -2 \end{pmatrix}$$

… And Computers Can Do It Too!

```
## create matrices
A = np.matrix([[1, 2, 0],
               [0, -1, 1],
               [1, -4, 1]])
B = np.matrix([[7], [6], [-1]])
## compute
A1 = np.linalg.inv(A)
x = A1.dot(B)
```

Another Example: Linear Regression

Non-matrix problem (2 variables):

$$Y_i = \beta_1 + \beta_2 T_i + \beta_3 G_i + \epsilon_i \qquad \text{for} \quad i = 1 \dots n$$

Inserting the specific variables of the example, we have

$$\begin{aligned}
b_1 n &+ b_2 \Sigma_i T_i &+ b_3 \Sigma_i G_i &= \Sigma_i Y_i, \\
b_1 \Sigma_i T_i &+ b_2 \Sigma_i T_i^2 &+ b_3 \Sigma_i T_i G_i &= \Sigma_i T_i Y_i, \\
b_1 \Sigma_i G_i &+ b_2 \Sigma_i T_i G_i &+ b_3 \Sigma_i G_i^2 &= \Sigma_i G_i Y_i.
\end{aligned}$$

A solution can be obtained by first dividing the first equation by n and rearranging it to obtain

$$\begin{aligned}
b_1 &= \bar{Y} - b_2 \bar{T} - b_3 \bar{G} \\
&= 0.20333 - b_2 \times 8 - b_3 \times 1.2873. \qquad \textbf{(3-7)}
\end{aligned}$$

a set of two equations:

$$\begin{aligned}
b_2 \Sigma_i (T_i - \bar{T})^2 &+ b_3 \Sigma_i (T_i - \bar{T})(G_i - \bar{G}) = \Sigma_i (T_i - \bar{T})(Y_i - \bar{Y}), \\
b_2 \Sigma_i (T_i - \bar{T})(G_i - \bar{G}) &+ b_3 \Sigma_i (G_i - \bar{G})^2 = \Sigma_i (G_i - \bar{G})(Y_i - \bar{Y}).
\end{aligned} \qquad \textbf{(3-8)}$$

This result shows the nature of the solution for the slopes, which can be computed from the sums of squares and cross products of the deviations of the variables. Letting lowercase letters indicate variables measured as deviations from the sample means, we find that the least squares solutions for b_2 and b_3 are

$$b_2 = \frac{\Sigma_i t_i y_i \Sigma_i g_i^2 - \Sigma_i g_i y_i \Sigma_i t_i g_i}{\Sigma_i t_i^2 \Sigma_i g_i^2 - (\Sigma_i g_i t_i)^2} = \frac{1.6040(0.359609) - 0.066196(9.82)}{280(0.359609) - (9.82)^2} = -0.0171984,$$

$$b_3 = \frac{\Sigma_i g_i y_i \Sigma_i t_i^2 - \Sigma_i t_i y_i \Sigma_i t_i g_i}{\Sigma_i t_i^2 \Sigma_i g_i^2 - (\Sigma_i g_i t_i)^2} = \frac{0.066196(280) - 1.6040(9.82)}{280(0.359609) - (9.82)^2} = 0.653723.$$

With these solutions in hand, the intercept can now be computed using (3-7); $b_1 = -0.500639$.

Matrix Approach

$$\begin{pmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_n \end{pmatrix} = \begin{pmatrix} 1 & T_1 & G_1 \\ 1 & 1_2 & G_2 \\ \vdots & \vdots & \vdots \\ 1 & T_n & G_n \end{pmatrix} \begin{pmatrix} \beta_1 \\ \beta_2 \\ \beta_3 \end{pmatrix} + \begin{pmatrix} \epsilon_1 \\ \epsilon_2 \\ \vdots \\ \epsilon_n \end{pmatrix}$$

Solution:

$$\begin{pmatrix} \beta_1 \\ \beta_2 \\ \beta_3 \end{pmatrix} = (XX')^{-1} X' \begin{pmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_n \end{pmatrix} \qquad \text{where} \qquad X = \begin{pmatrix} 1 & T_1 & G_1 \\ 1 & 1_2 & G_2 \\ \vdots & \vdots & \vdots \\ 1 & T_n & G_n \end{pmatrix}$$

```
## warning: not tested
np.linalg.inv(X.dot(X.T)).dot(X.T).dot(y)
```

Summary

- Linear algebra is a very important language in statistics
  - A lot of multivariate stuff easily generalizes into matrix form
  - Good software libraries exist
  - A lot less error prone
  - You should be able to read it ...
  - ...and code it

Matrix

$$A = [a_{ij}] = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1M} \\ a_{21} & a_{22} & \dots & a_{2M} \\ \vdots & \vdots & \ddots & \vdots \\ a_{N1} & a_{N2} & \dots & a_{NM} \end{bmatrix}$$

Symmetric matrix $a_{ij} = a_{ji}$

Diagonal Matrix $a_{ij} = 0$ iff $i \neq j$

Identity matrix $a_{ij} = \mathbb{1}(i = j)$

Transposition

$$A' = [a_{ji}] = \begin{bmatrix} a_{11} & a_{21} & \dots & a_{N1} \\ a_{12} & a_{22} & \dots & a_{N2} \\ \vdots & \vdots & \ddots & \vdots \\ a_{1M} & a_{2M} & \dots & a_{NM} \end{bmatrix}$$

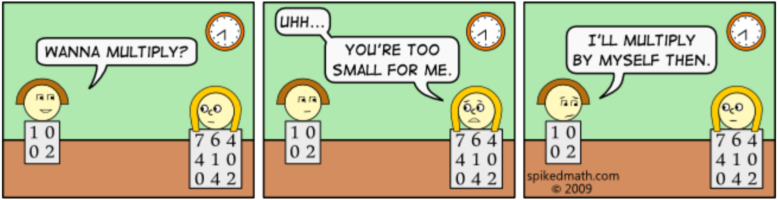
Matrix Multiplication

$$AB = [c_{ik}]$$

where

$$c_{ik} = \sum_{j=1}^{M} a_{ij} b_{jk}$$

- Number of rows of A must match the number of columns of B



Matrix Multiplication

- Properties:
  - Associative: $(AB)C = A(BC)$
  - Distributive: $A(B + C) = AB + AC$
  - Not commutative: $(AB)' = B'A'$ but $AB \neq BA$
- *Left-multiplication* and *right-multiplication*

Vector

$1 \times n$ or $n \times 1$ matrix:

- Column vector

$$\mathbf{a} = \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{bmatrix}$$

- Row vector

$$\mathbf{a}' = \begin{bmatrix} a_1 & a_2 & \dots & a_n \end{bmatrix}$$

- Vector multiplication is regular matrix multiplication

$$aa' = [a_i a_j] \qquad \text{is } n \times n \text{ matrix}$$

$$a'a = \left[\sum_{i=1}^{n} a_i^2\right] \qquad \text{is } 1 \times 1 \text{ matrix (scalar)}$$

Linear Independence

Vectors $\mathbf{a}_1$, $\mathbf{a}_2$, ..., $\mathbf{a}_n$ are *linearly independent* iff

$$\alpha_1 \mathbf{a}_1 + \alpha_2 \mathbf{a}_2 + \cdots + \alpha_n \mathbf{a}_n = 0$$
$$\Updownarrow$$
$$\alpha_1 = \alpha_2 = \cdots = \alpha_n = 0$$

Column Space, Rank

Column space is the vector space, generated by the column vectors of the matrix

Column rank dimension of the column space

Full column rank column rank equals to the number of columns

Full rank rank equals to the smallest of either number of rows or number of columns

Theorem

Row and column rank are equal

(and are called *matrix rank*)

Determinant

is a scalar function of matrix elements

- Useful descriptor of matrix properties in many contexts
- For 2×2 matrix $A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}$

$$\det A \equiv |A| = a_{11}a_{22} - a_{12}a_{21}$$

- for 3×3 matrix $B = \begin{bmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \\ b_{31} & b_{32} & b_{33} \end{bmatrix}$

$$\begin{aligned} |B| = b_{11}b_{22}b_{33} + b_{12}b_{23}b_{31} + b_{21}b_{32}b_{13} - \\ - b_{31}b_{22}b_{13} - b_{21}b_{12}b_{33} - b_{11}b_{23}b_{32} \end{aligned}$$

Determinant 2

- Determinant of diagonal matrix

$$C = \begin{bmatrix} c_{11} & 0 & \dots & 0 \\ 0 & c_{22} & \dots & 0 \\ \dots & \dots & \ddots & \vdots \\ 0 & 0 & \dots & c_{nn} \end{bmatrix}$$

$$|C| = \prod_{i=1}^{n} c_{ii}$$

Determinant Property

Theorem

determinant is non-zero $\Leftrightarrow$ *matrix is full rank*

Inverse Matrix

- A: square matrix

B is *inverse* A iff

$$BA = I,$$

and is denoted by A^{-1}.

- A^{-1} is also a square matrix
- A^{-1} is unique

For 2×2 matrix $A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}$,

$$A^{-1} = \frac{1}{|A|} \begin{bmatrix} a_{22} & -a_{12} \\ -a_{21} & a11 \end{bmatrix}$$

Inverse Matrix 2

- Inverse of diagonal matrix $C = \begin{bmatrix} c_{11} & 0 & \dots & 0 \\ 0 & c_{22} & \dots & 0 \\ \dots & \dots & \ddots & \vdots \\ 0 & 0 & \dots & c_{nn} \end{bmatrix}$

$$C^{-1} \begin{bmatrix} 1/c_{11} & 0 & \dots & 0 \\ 0 & 1/c_{22} & \dots & 0 \\ \dots & \dots & \ddots & \vdots \\ 0 & 0 & \dots & 1/c_{nn} \end{bmatrix}$$

Inverse Properties

Definition

Matrix is *non-singular* $\Leftrightarrow$ inverse exists

Theorem

Matrix is non-singular $\Leftrightarrow$ *it is full rank*

Characteristic Equation

Characteristic roots (eigenvalues) are solutions (λ) of the equation

$$\mathsf{A}\mathbf{c} = \lambda\mathbf{c}$$

Characteristic vectors (eigenvectors) are corresponding $\mathbf{c}$-s. Rewrite:

$$(\mathsf{A} - \lambda\mathsf{I})\mathbf{c} = 0$$

$\Rightarrow \mathsf{A} - \lambda\mathsf{I}$ must be singular $\Rightarrow |\mathsf{A} - \lambda\mathsf{I}| = 0$.

Symmetric Matrix

$n \times n$ symmetric matrix has

- n distinct characteristic vectors $\mathbf{c}_i$
- n real characteristic roots (not necessarily distinct)
- Characteristic vectors are orthogonal:

$$\mathbf{c}_i'\mathbf{c}_j = \mathbb{1}(i = j)$$

Diagonalization

Characteristic equation in matrix form

$$AC = C\Lambda,$$

where

$$C = \begin{bmatrix} \mathbf{c}_1 & \mathbf{c}_2 & \dots & \mathbf{c}_n \end{bmatrix}$$ and

$$\Lambda = \begin{bmatrix} \lambda_0 & 0 & 0 & \dots & 0 \\ 0 & \lambda_1 & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots & 0 \\ 0 & 0 & 0 & \dots & \lambda_n \end{bmatrix}$$

Characteristic Roots of Symmetric Matrix

As characteristic vectors $\mathbf{c}$ orthogonal,

$$\mathsf{C}'\mathsf{C} = \mathsf{I}$$

- C is *idempotent* matrix
- $\mathsf{C}^{-1} = \mathsf{C}'$

Diagonalization:

$$\mathsf{A} = \mathsf{C}\Lambda\mathsf{C}'$$

or

$$\Lambda = \mathsf{C}'\mathsf{A}\mathsf{C}$$

Rank

Theorem

Rank of a symmetric matrix equals the number of non-zero characteristic roots

$$\operatorname{rank} A = \operatorname{rank} \Lambda$$

Condition Number

Show how close a matrix is to being singular:

$$\kappa(A) = \frac{|\lambda_{max} A|}{|\lambda_{min} A|}$$

- λ_{max} maximum eigenvalue (by absolute value)
- λ_{min} minimum
- Shows the sensitivity (to numeric noise/errors
- Sometimes defined as $\sqrt{\cdot}$ of the ratio above