Matrix Algebra

Jacob Denson

December 19, 2023

Table Of Contents

Chapter 1

Basic Definitions

Chapter 2

Linear Equations

The fundamental problem in linear algebra is to solve a system of m equations in n variables. One might write this system as

$$\sum a_{1j}x_j = b_1 \quad \dots \quad \sum a_{mj}x_j = b_m.$$

The quantities $\{a_{ij}\}$ are known, the *coefficients* of the equation, as are the quantities $\{b_j\}$, and the goal is to figure out the value of the unknown values $\{b_j\}$. The tricks we describe to solve this equation were known in China, and later Japan, from the years 200BC. But they were formalized in Europe by Carl Gauss in the 19th century, and so the technique is called *Gaussian elimination*. To make the calculus of Gaussian elimination more easy to see, we will write these equations in matrix form. If A is the $m \times n$ matrix with coefficients $\{a_{ij}\}$, $\mathbf{b}$ is the column vector with entries $\{b_j\}$, and $\mathbf{x}$ is the column vector of unknowns $\{x_i\}$, then we can write the equation as $A\mathbf{x} = \mathbf{b}$.

The trick to Gaussian elimination is to successively write these equations in *equivalent ways* that are successively more simple than one another. There are three *elementary operations* one can perform, that can easily be seen to lead to an equivalent system of equations:

- Swapping one equation with another equation.
- Multiplying both sides of an equation by a non-zero number.
- Adding one equation to another equation.

We claim that these three rules can be used to put these equations in a simplified form in which it is easy to determine solutions of the equation.

At each step of the algorithm, we identify a *pivot column*, and identify an equation

Part I

Computational Linear Algebra

2.1 Finding Eigenvalues

Let A be an $n \times n$ matrix. The classical method of finding eigenvalues of A is to consider the roots of the characteristic polynomial

$$p(\lambda) = \det(A - \lambda).$$

The polynomial p has degree n, which makes it difficult to find roots when n is large. There are several iterative methods to find the eigenvalues of A. Let us make the assumption that A is diagonalizable, which is valid in practice since this is true of a generic matrix. Let $\lambda_1, \dots, \lambda_n$ denote the eigenvalues, repeated up to multiplicity and ordered in such a way that $|\lambda_1| \geqslant \cdots \geqslant |\lambda_n|$, and consider a normalized eigenbasis $e_1, \dots, e_n$. Consider some normalized vector

$$x = a_1 e_1 + \cdots + a_n e_n.$$

Then

$$A^N x = \lambda_1^N a_1 e_1 + \cdots + \lambda_n^N a_n e_n.$$

If $|\lambda_j| \leqslant \delta |\lambda_1|$ for some $0 < \delta < 1$ and $j \neq 2$, then

$$\left\| \frac{A^N x}{\lambda_1^N} - a_1 e_1 \right\| \leqslant \sum_{j=2}^{n} |a_j| \frac{|\lambda_j|^N}{|\lambda_1|^N} \leqslant \delta^N \sum_{j=2}^{n} |a_j| \lesssim \delta^N,$$

and so one can compute approximate eigenvectors with eigenvalue λ_1 very quickly. We can then approximate the eigenvalue λ_1, provided that a_1 is not chosen too small. If $|a_1| \geqslant \varepsilon$, and if we set $y = A^N x$, then $\|y\| = |\lambda_1|^N a_1 + O(\delta^N)$, and so

$$\frac{\langle Ay, y\rangle}{\langle y, y\rangle} = \frac{\lambda_1^{2N+1}|a_1|^2 + \lambda_2^{2N+1}|a_2|^2 + \cdots + \lambda_n^{2N+1}|a_n|^2}{\lambda_1^{2N}|a_1|^2 + O(\delta^N)} = \lambda_1 \left(1 + O(\delta^N \lambda_1^{-2N} \varepsilon^{-2})\right)$$