Matrix Theory and Applications, Day 1: Fundamentals

Sarah Marzen

February 20, 2013

Linear algebra essentially consists of the following setup

$$a_{i1}x_1 + a_{i2}x_2 + ... + a_{in}x_n = b_i,\ \ i = 1, ..., n \tag{1}$$

Now we want to solve for $\mathbf{x} = (x_1, ..., x_n)$. Can be re-expressed in the following form:

$$A\mathbf{x} = \mathbf{b} \tag{2}$$

where $A = [a_{ij}]_{i,j=1}^n$, $\mathbf{b} = (b_1, ..., b_n)$.

Note: why don't we have nonlinear equations, with squares, instead of just linear ones? Just one step above linear, you get a quadratic problem, for instance. Hilbert thought we could solve those easily, but Matiyasevich proved that in general, there is no computer that can take as input a nonlinear problem and output a solution. However, you can use the framework of linear algebra to study nonlinear systems. For instance, if you want to minimize the quadratic $\mathbf{x}^\top A\mathbf{x}$, you can use gradient descent to get a linear system of equations, even though your initial problem is nonlinear (quadratic).

1 A sampling of problems that can be solved using matrix theory and linear algebra

What problems might you want to solve using linear algebra? This is just a sampling of the problems covered in this course.

1.1 Interpolation/Data fitting

What is the next number in this sequence: $1, 1, 2, 3, 5, ?$ The question mark can be anything!! You can find a polynomial that will give you any value for the next number in the series.

More precisely, suppose you are given data $x_1, x_2, ..., x_n$ the independent variable and $y_1, ..., y_n$ the dependent variable. Generally you want to find a polynomial $p(x)$ such that $p(x_i) = y_i\ \forall i = 1, ..., n$. It turns out that $p(x_{n+1})$ is completely unconstrained.

For instance, suppose that you have $p(x) = c_3x^3 + c_2x^2 + c_1x + c_0$; this is a "model" for our "data". We want to find c_0, c_1, c_2, c_3, the parameters of our model, such that the polynomial gives a best fit to our data: $y_1 = p(x_1)$, $y_2 = p(x_2)$, $y_3 = p(x_3)$ and $y_4 = p(x_4)$. Let's write down what this actually looks like:

$$\begin{pmatrix} x_1^3 & x_1^2 & x_1 & 1 \\ x_2^3 & x_2^2 & x_2 & 1 \\ x_3^3 & x_3^2 & x_3 & 1 \\ x_4^3 & x_4^2 & x_4 & 1 \end{pmatrix} \begin{pmatrix} c_3 \\ c_2 \\ c_1 \\ c_0 \end{pmatrix} = \begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{pmatrix} \tag{3}$$

So in matrix form, this is

$$Xc = y \tag{4}$$

where we are trying to find **c**. So let's multiply both sides of this equation by X^{-1} to find c:

$$c = X^{-1}y \tag{5}$$

You might ask, hey, sometimes X^{-1} does not actually exist! In fact, X^{-1} exists if and only if $\det X \neq 0$. So can we actually invert this X? This determinant, called the *Vandermonde determinant*, satisfies:

$$\det X = (x_1 - x_2)(x_1 - x_3)(x_1 - x_4)(x_2 - x_4)(x_2 - x_3)(x_3 - x_4) \tag{6}$$

and this is generalizable to larger degree polynomials. So in general if you have $X_{ij} = x_i^{j-1}$, then

$$\det X = \prod_{i<j}(x_i - x_j) \tag{7}$$

Proof: left to the reader as an exercise. Try induction, row/column reductions! So as long as you don't have two different y's for the same x, you're good to go.

1.2 Fibonacci numbers

This famous sequence was originally invented to describe the number of rabbits in a population of breeding rabbits. Suppose that the number of offspring at time $n+2$, f_{n+2}, is given by

$$f_{n+2} = f_{n+1} + f_n \tag{8}$$

with $f_1 = 1$ and $f_2 = 1$. This completely determines the sequence $\{f_1, f_2, f_3, ..., f_n, ..\}$. So what does this sequence look like? Recursive sequences are generally hard to understand. So we're going to turn this into linear algebra.

Claim: This sequence f can be generated as follows.

$$\begin{pmatrix} f_{n+2} \\ f_{n+1} \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}^n \begin{pmatrix} f_2 \\ f_1 \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}^n \begin{pmatrix} 1 \\ 1 \end{pmatrix}. \tag{9}$$

Why is this true? Try an exercise with $n = 1$, but the general proof is possible by induction. *Proof: left to the reader as an exercise. Try induction!*

We still don't understand the system, though we now have it in the form

$$\begin{pmatrix} f_{n+2} \\ f_{n+1} \end{pmatrix} = A^n \begin{pmatrix} 1 \\ 1 \end{pmatrix}. \tag{10}$$

where we can write A in a very useful form:

$$A = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix} = SDS^{-1} \tag{11}$$

Here

$$D = \begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix} \tag{12}$$

is a diagonal matrix of eigenvalues and S is a matrix of eigenvectors. Now

$$A^n = (SDS^{-1})^n = SD^nS^{-1} = S \begin{pmatrix} \lambda_1^n & 0 \\ 0 & \lambda_2^n \end{pmatrix} S^{-1}. \tag{13}$$

With these ingredients, you can show that for some α, β, we have for all n:

$$f_{n+2} = \alpha\lambda_1^n + \beta\lambda_2^n, \tag{14}$$

and the proof of this is left as an exercise to those readers that already know what eigenvalues are.

1.3 Principal Components Analysis

Here is one way to interpret PCA. Given a random vector $\begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix}$ (with mean 0, say) find a linear transformation T such that a transformed vector y

$$y = \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix} = Tx \tag{15}$$

has that the elements of y are uncorrelated. This is sometimes called whitening. More precisely,

$$\mathrm{Cov}(y) = \mathbb{E}(yy^\top) \tag{16}$$

should be a diagonal matrix, i.e. should have nonzero elements only on its diagonal. $\mathbb{E}$ refers to the expectation value, as x and thus y is a random variable. To be very clear, $\mathrm{Cov}(y)$ is the covariance matrix of size n by n. By the way, $yy^\top$ is called an *outer product.*

So how do we do this?

$$\mathbb{E}(yy^\top) \quad = \quad \mathbb{E}((Tx)(Tx)^\top) = \mathbb{E}(Txx^\top T^\top) = T\mathbb{E}(xx^\top)T^\top. \tag{17}$$

If C is the covariance of x, then we are looking for a matrix T such that

$$I = TCT^\top. \tag{18}$$

This ends up meaning that you want to diagonalize C. In fact, a common solution is to take $T = C^{-1/2}$ when C is invertible, which requires computing a special diagonalization (or SVD) of C.

1.4 Data analysis (e.g., Netflix challenge)

Given some big data matrix A that has some underlying low-rank structure, for instance $A = yx^\top + \eta$ where x, y are some vectors and η is some noise, how do you recover the low-rank structure of A? For instance, if A is movie preferences, x could be the identity of people and y the identity of movies.

Suppose that we think that the rank of the underlying data is r. Then we want to find the rank r matrix X that minimizes the Froebenius (or some other norm) between your data A and X, i.e. find $X = \arg\min_{\mathrm{rank}(X)=r} ||A - X||_2$. Singular value decomposition solves this: just take the r largest singular values in the singular value decomposition of A and set the rest to 0.

1.5 Machine learning: Image segmentation

Check out reference (Shi, Malik 2000), but basically, how do you segment an image? We're going to turn the problem into a combinatorial optimization and then relax the combinatorial optimization to finding eigenvectors of a matrix. This method, in general, of relaxing a combinatorial optimization is a spectral method. Here is a sketch of how this works for image segmentation:

- Take your image, turn it into a vector with elements that are the pixel intensities.
- Compute a "similarity matrix" A which has size number of pixels in image by number of pixels in image such that $A_{xy} = e^{-|I(x)-I(y)|}e^{-\mathrm{dist}(x,y)}$, where $\mathrm{dist}(x, y)$ is the distance between the pixels x and y in the image and $I(x) - I(y)$ is the difference between the intensities of pixels x and y. Sometimes you subtract off the identity (for some reason that I didn't catch)

- You can view this similarity matrix as a weighted graph in which the vertices aka nodes are pixels and the weights on the edges between the nodes are given by the elements of the similarity matrix. So "similar" pixels will have stronger connections than less similar pixels. In this view, segments are equivalent to large highly connected subgraphs, meaning that pixels from the same image are more similar.
- By magic, you can turn the problem of finding highly connected subgraphs into a combinatorial optimization problem, which involves minimizing a quadratic form, $\arg\min_{x\in\{-1,1\}^n} x^\top M(A)x$. More on that later in the course. $M(A)$ is just some matrix function of A which happens to be the Laplacian of A. Unfortunately this problem is NP-hard; the possible solutions are solutions on an n dimensional hypercube, and to actually solve this problem, you'd need to test all the possible vertices of the hypercube. If you can solve this faster than exponential, you win a million dollars, literally.
- So we can pretend that x's elements are real numbers between $[-1, 1]$, and this is not an NP-hard problem; the solution is that we want the eigenvector of $M(A)$ corresponding to the smallest eigenvalue of $M(A)$.

This is how Google got famous, somehow, since if you only need a certain eigenvector or eigenvalue you can sidestep a lot of computational difficulties. Why on earth would this work? An eigenvector is sort of a global computation of a matrix that you need to see the entire matrix for.

2 Review of the fundamentals of linear algebra

Loosely a vector space has the following properties:

- You can scale objects.
- You can add them.

This excludes some properties: you cannot multiply objects and stay inside the vector space (think the cross-product of two vectors that takes you from $\mathbb{R}^3$ to $\mathbb{R}^{3\times 3}$), at least not necessarily. The mathematical version of this is as follows– we do it for vector spaces over $\mathbb{R}$ but you can easily extend this to other fields like $\mathbb{C}$.

——

Def'n of a vector space: A vector space over $\mathbb{R}$ is a set V equipped with a scalar multiplication, denoted as $\cdot$, and a vector addition, denoted as $+$. The vector multiplication $\cdot$ maps $\mathbb{R} \times V$ to V, and we write that as $\cdot : \mathbb{R} \times V \to V$, written element wise as $\cdot : (r, v) \to r \cdot v$. Multiplication has the following properties:

- distributive?, $a \cdot (u + v) = a \cdot u + a \cdot v$
- distributive again?, $(a + b) \cdot u = a \cdot u + b \cdot u$
- associative?, $a \cdot (bu) = (ab) \cdot u$
- identity, $1 \cdot u = u$.

Addition "forms an abelian group", explicitly meaning that it is an operation that takes elements of $V \times V$ to V, written elementwise as $+ : (u, v) \to u + v$. Addition has the following properties:

- commutative, $u + v = v + u$
- associative, $u + (v + w) = (u + v) + w$
- identity 0 exists, $u + 0 = u$
- inverse exists $-u$, $u + (-u) = (-u) + u = 0$

—–

Examples of vector spaces– proofs are left as exercises to the reader:

- Elements of $\mathbb{R}^n$, which are the vectors that we usually deal with
- Polynomials of degree n, with an element $a_0 + a_1 t + a_2 t^2 + ... + a_n t^n$ with coefficients $a_i \in \mathbb{R}$
- Continuous functions $f : [0, 1] \to \mathbb{R}$. Prove that discontinuous functions are NOT a vector space.
- Matrices.

What are examples of NOT a vector space? Look at the upper right quadrant of the normal $\mathbb{R}^2$ vector space, the Cartesian coordinate plane. This has all of the properties of a vector space *except* that it does not have an inverse.

Def'n of linear transformation: Linear transformation L takes elements of a vector space V to elements of V, with the property that $L(au+bv) = aL(u)+bL(v)$ where $a, b \in \mathbb{R}$ and $u, v \in V$. For instance, L could be a matrix applied to a vector when V is $\mathbb{R}^n$ or a Fourier transform if V is the vector space of continuous functions.

Def'n of subspace: Subspace W in a vector space V is contained in V and is also a vector space. This is equivalent to the statement that $aw_1 + bw_2 \in W$ for any $w_1, w_2 \in W$ and any $a, b \in \mathbb{R}$. Proof left to the reader; it follows from the definition of vector space. For instance, all even degree polynomials is a subspace of polynomials and the vectors $(x_1, ..., x_n)$ in $\mathbb{R}^n$ with $x_1 + x_2 + ... + x_n = 0$ is a subspace of $\mathbb{R}^n$.

Def'n of linearly dependent: $u_1, ..., u_n \in V$ are linearly dependent if there exists numbers $c_1, ..., c_n \in \mathbb{R}$ with at least one of $c_i \neq 0$ such that $c_1 u_1 + ... + c_n u_n = 0$ (i.e., we can write one of these vectors as linear combinations of the others).

Def'n of linearly independent: $u_1, ..., u_n \in V$ are linearly independent if they are not linearly dependent, i.e. that $c_1 u_1 + ... + c_n u_n = 0 \Leftrightarrow c_i = 0 \ \forall \ i = 1, ..., n$. Example: the set $\{1, t, ..., t^n\}$ are linearly independent.

Def'n of span: The span of $u_1, ..., u_n$ is the set of all the linear combinations of those vectors,

$$\text{span}\{u_1, ..., u_n\} = \{\sum_{i=1}^{n} c_i u_i : c_i \in \mathbb{R}\} \tag{19}$$

Def'n of bases: a basis B for a vector space V is a set of elements $B = \{b_1, ..., b_n\}$ such that two properties hold–

- $\text{span}(B) = V$, so you get everything in V from B.
- $b_1, ..., b_n$ are linearly independent, so there's no redundancy.

Theorem: (a) Every vector space V has a basis and (b) the cardinality[1] of every basis of V is the same and this size is called the "dimension" of V, $\dim(V)$. *Exercise*: Prove this theorem for $V = \mathbb{R}^n$.

Examples: $\dim(\mathbb{R}^n) = n$.

Homework: let $W = \{(x_1, ..., x_n) : \sum_{i=1}^{n} x_i = 0\}$; what is $\dim(W)$? Give a basis for W.

[1] If finite, the cardinality is just the number of elements in the set; things get complicated with infinities