EE103 Prof. S. Boyd

Mathematical Notation Versus `Julia` Syntax

In the tables below we show how to express some mathematical notation (as in the textbook *Vectors, Matrices, and Least Squares*) in the computer language `Julia`. Be careful to never confuse mathematical notation and `Julia` syntax!

In the tables below we use `this font` to denote things you'd type in to `Julia`.

Vectors

Basics

concept	mathematical notation	`Julia` syntax
n-vector	$(x_1, \ldots, x_n)$, or in column format, $\begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}$ or $\begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix}$.	Represented as `1-d array` of length `n`. For example, a 3-vector can be written as `[x_1, x_2, x_3]` or `[x_1; x_2; x_3]`. If you type `x` in interactive mode, it will be displayed as a column.
vector entries	x_i.	`x[i]`.
vector size	n (x has n entries).	`length(x)`.
vector slice	$x_{i:j} = (x_i, \ldots, x_j)$.	`x[i:j]`.
stacking	$(x, y) = (x_1, \ldots, x_n, y_1, \ldots, y_m)$	`[x; y]`.
equality	$x = y$.	`x==y` returns `true` or `false`. (`x=y` assigns `x` to the value of `y`.)

list of vectors	$x_1, \ldots, x_k$.
	x_i: the ith vector.
	$(x_i)_j$: jth entry of x_i.
		`# list of vectors`
		`list = [x_1, x_2, x_3]`
		`# first vector`
		`list[1]`
		`# third entry of second vector`
		`list[2][3]`

Specific vectors

concept	mathematical notation	Julia syntax
zero vector	0_n or (more commonly) just 0.	`zeros(n)`.
ones vector	$\mathbf{1}_n$ or $\mathbf{1}$.	`ones(n)`.

unit vectors	$e_i = (0, \ldots, 0, 1, 0, \ldots, 0)$ (ith entry is one).	No built-in `Julia` syntax for unit vectors. The following code creates e_i:
		`# create zero vector`
		`ei = zeros(n)`
		`# set i-th entry to 1`
		`ei[i] = 1`

Vector operations and functions

In the table below we give the native `Julia` syntax, and the syntax using a simple module called `MMA`, which contains `Julia` definitions of some common functions arising in the course.

concept	mathematical notation	Julia syntax
vector addition, difference	$x + y$, $x - y$.	`x + y`, `x - y`.
scalar-vector multiplication	ax (or xa), with a a number.	`a*x` or `x*a`.
vector sum	$\mathbf{1}^T x$.	`sum(x)`.
scalar-vector addition	$x + a\mathbf{1}$.	`x .+ a` or `a .+ x`.
inner product	$x^T y$.	`dot(x, y)`.
vector norm	$\|x\|$.	`norm(x)`.
RMS value	$\mathbf{rms}(x) = \|x\|/\sqrt{n}$.	`norm(x)/sqrt(length(x))`. Using `MMA`: `rms(x)`.
distance	$\mathbf{dist}(x, y) = \|x - y\|$.	`norm(x-y)`. Using `MMA`: `dist(x, y)`.
average	$\mathbf{avg}(x) = (x_1 + \cdots + x_n)/n$.	`mean(x)`.

de-mean	$x - \mathbf{avg}(x)\mathbf{1}$.	`x - mean(x)`. Using `MMA`: `demean(x)`.
standard deviation	$\mathbf{std}(x)$.	`norm(x-mean(x))/sqrt(length(x))`. Using `MMA`: `std(x)`.
angle	$\angle(x, y)$.	`acos(dot(x,y)/(norm(x)*norm(y))`. Using `MMA`: `angle(x, y)`.

correlation coefficient	$\rho(x, y)$.	No built-in function for correlation coefficient. The following code computes it:
		`# de-mean vectors`
		`xt = x-mean(x); yt = y-mean(y)`
		`rho = dot(xt,yt)/(norm(xt)*norm(yt))`.
		Using `MMA`: `corrcoef(x, y)`.

convolution	$x * y$	`conv(x,y)`.

Matrices

Basics

concept	mathematical notation	Julia syntax

$m \times n$ matrix	$A = \begin{bmatrix} A_{11} & \cdots & A_{1n} \\ \vdots & \ddots & \vdots \\ A_{m1} & \cdots & A_{mn} \end{bmatrix}$.	Represented as `2-d array` of size `m` × `n`. For example, a 2×3 matrix can be written as
		`A = [A_11, A_12, A_13;`
		`     A_21, A_22, A_23].`
		Typing `A` in interactive mode displays the entries of A.

matrix entries	A_{ij}.	`A[i,j]`.
matrix dimensions	$m \times n$.	`m, n = size(A)`. To get row or column dimensions separately: `m = size(A)[1]` `n = size(A)[2]`.
submatrices	$A_{p:q,r:s} = \begin{bmatrix} A_{pr} & A_{p,r+1} & \cdots & A_{ps} \\ A_{p+1,r} & A_{p+1,r+1} & \cdots & A_{p+1,s} \\ \vdots & \vdots & & \vdots \\ A_{qr} & A_{q,r+1} & \cdots & A_{qs} \end{bmatrix}$.	`A[p:q, r:s]`.
block matrix	A $= \begin{bmatrix} B & C \\ D & E \end{bmatrix}$.	`A = [B C; D E]`.
equality	$A = B$.	`A==B` returns `true` or `false`. (`A = B` assigns `A` to the value of `B`.)

Specific matrices

concept	mathematical notation	`Julia` syntax
zero matrix	$0_{m\times n}$ or, more commonly, 0.	`zeros(m,n)`.
identity matrix	$I_{n\times n}$ or, more commonly, I.	`eye(n)`

Matrix operations and functions

concept	mathematical notation	`Julia` syntax
matrix transpose	A^T.	`A'` or `transpose(A)`.
matrix-matrix sum, difference	$A+B$, $A-B$.	`A + B`, `A - B`.
column selection	jth column of A.	`A[:,j]`.
row selection	jth row of A.	`A[j,:]`.
scalar-matrix product	bA (or Ab), with b a number.	`b*A` or `A*b`.
matrix-vector product	Ax (A an $m\times n$ matrix, x an n-vector).	`A*x`.
matrix-matrix product	AB (A an $m\times n$ matrix, B an $n\times p$ matrix).	`A*B`.
matrix power	A^k (A square, k integer ≥ 1).	`A^k`.
matrix inverse	A^{-1} (A square, invertible).	`inv(A)`.
matrix pseudo-inverse	$A^\dagger$.	`pinv(A)`.
diagonal matrix	$\mathbf{diag}(d)$, with d a vector	`diagm(d)`.

Linear equations and least squares

concept	mathematical notation	`Julia` syntax

solve equations	$x = A^{-1}b$ (A invertible)	`x=A\b`.
least squares	$x = (A^TA)^{-1}A^Tb$ (A has independent columns)	`x=A\b`.
least-norm	$x = A^T(AA^T)^{-1}b$ (A has independent rows)	`x=A\b`.