Class—FCLA JCF

Advanced Linear Algebra

Robert Beezer

Math 390, Spring 2021

The Jordan Canonical Form of a matrix is basically a combinatorial computation, once we have the eigenvalues of the matrix (which is a hard computation). Here we will find a basis for the matrix representation first, then show how to find the representation only (without finding a basis first), by just knowing dimensions of the kernels of powers of matrices formed from eigenvalues.

We begin with a large-ish matrix in order to demonstrate the most general situations.

```
A = matrix(QQ,
[[50, -10, 17, -21, 7, 6, 1, -37, -8, -10, 10, -14, 27, 39,
    8],
[186, -207, -36, -184, -74, -71, -54, -22, -1, -93, 38, -11,
    56, 187, 14],
[3, 16, 9, -10, -11, 8, -4, -24, 4, -6, 2, -14, 13, 6, -2],
[-132, 202, 62, 162, 83, 77, 58, -23, -14, 86, -28, -4, -22,
    -149, -3],
[213, -237, -41, -218, -94, -82, -65, -30, 2, -111, 45, -16,
    69, 216, 13],
[-608, 549, 27, 552, 195, 173, 139, 178, 34, 266, -124, 72,
    -239, -574, -57],
[253, -299, -57, -274, -125, -108, -83, -26, 3, -137, 52,
    -15, 79, 258, 15],
[-196, 175, 0, 170, 50, 56, 38, 58, 21, 76, -38, 18, -76,
    -176, -23],
[704, -651, -40, -649, -233, -206, -167, -197, -35, -313,
    143, -80, 272, 668, 64],
[322, -238, 25, -251, -53, -62, -51, -126, -39, -112, 62,
    -43, 139, 283, 42],
[-536, 411, -28, 449, 129, 115, 98, 211, 52, 207, -105, 79,
    -236, -482, -61],
[170, -190, -25, -165, -58, -67, -45, -20, -11, -77, 32, -4,
    51, 163, 17],
[1, 14, 9, 10, 9, 7, 6, -7, -4, 5, 0, -1, 7, -4, 1],
[223, -195, -1, -201, -64, -61, -49, -71, -20, -92, 43, -26,
    91, 206, 25],
[-58, 4, -27, 29, -7, -10, 1, 53, 12, 12, -11, 20, -39, -45,
    -10]
])
A
```

Eigenvalues first, a nontrivial computation. In this example, they are all integers, which is atypical.

```
A.eigenvalues()
```

We will work with $\lambda = 2$, leaving $\lambda = -1$ for you to experiment with. The the (invariant) generalized eigenspace for $\lambda = 2$ has dimension 10, so algebraic multiplicity is $\alpha_A(2) = 10$.

```
((A-2)^15).nullity()
```

Converted to a linear transformation $A - 2I_{15}$ is nilpotent when restricted to the generalized eigenspace. We do not know the index yet, but we know we only have to look as far as the tenth power.

```
[((A-2)^i).nullity() for i in range(11)]
```

So the kernels of the powers of $A - 2I_{15}$ "top out" at the fifth power, so the index is $\iota_A(2) = 5$.

We now build a basis of the dimension 10 generalized eigenspace. We begin with vectors in the kernel of the fifth power, multiplying them by $A - 2I_{15}$ to move them down into kernels of progressively smaller powers. Properly ordered, we get a "nice looking" matrix representation.

Table 1 Basis Vectors in Kernels

Kernels	$N\left((A-2I_{15})^i\right)$	$i=1$	$i=2$	$i=3$	$i=4$	$i=5$
Dimension		4	7	8	9	10
		x_1	x_2	x_3	x_4	x_5
		x_6	x_7			
		x_8	x_9			
		x_{10}				

How much information did we have about the Jordan blocks owing to this eigenvalue, prior to actually computing the basis? The nullity of $A - 2I_{15}$ is $n(A - 2I_{15}) = 4$, so there will be 4 Jordan chains, each ending in a traditional eigenvector, associated with one Jordan block. So there are four blocks. With an index of 5, the lonest chain has 5 vectors, and so the largest block will be a 5×5 Jordan block.

To determine all of the sizes of the blocks, we compute the differences between the dimensions of the kernels. These are the s_i, $1 \leq i \leq p$ in the proof of Jordan canonical form for nilpotent matrices.

```
dims = [((A-2)^i).nullity() for i in range(6)]
dimdiffs = [dims[i+1]-dims[i] for i in range(5)]
dimdiffs
```

From these numbers we can deduce the construction of the basis vectors for the kernels of the powers of $A - 2I_{15}$, where we would begin with the kernel of the fifth power. The **Ferrers Diagram** of this **partition** of 10 is a display of the *transpose* of the basis vectors listed at the end of the proof of Jordan canonical form for nilpotent matrices.

```
P = Partition(dimdiffs)
print P
print P.ferrers_diagram()
```

We want to reflect this diagram.

```
flipped = P.conjugate()
print flipped
print flipped.ferrers_diagram()
```

So the length of each row is the size of the blocks (a sequence of basis vectors) and we can get this information directly from the list that is the “flipped” partition. So we build the blocks (not forgetting $\lambda = 2$!)

```
J = block_diagonal_matrix([jordan_block(2,5),
    jordan_block(2,2), jordan_block(2,2), jordan_block(2,
    1)])
J
```

Let’s check. Notice that this is one of the few matrix decompositions in Sage which does not automatically provide the similarity transformation. As the work above shows, we can determine the canonical form without ever computing a basis vector, we need only get nullities by counting non-pivot columns of matrices in reduced row echelon form (of powers of matrices).

```
A.jordan_form(transformation=False)
```

Sage will compute the similarity transformation, if asked.

```
A.jordan_form(transformation=True)
```

We can also provide a different similarity transformation, since we reverse-engineered this example (though our eigenvalues are in the wrong order).

```
S = matrix(QQ,
    [[1, 0, 0, 0, 0, 0, 0, 1, 0, -1, -2, -2, -1, 2, 4],
    [1, 1, 0, 2, 0, 2, -3, -2, -1, 1, 1, -3, -2, 2, 3],
    [0, 0, 1, 0, 0, 0, 0, 1, 0, 1, 0, -1, -5, 5, -3],
    [-1, -2, 0, -3, 0, -2, 4, 3, 1, -3, -3, 3, 3, -3, -1],
    [0, 1, 0, 1, 1, 2, -2, -1, 0, 2, 2, -3, -2, 1, 1],
    [0, 0, 0, 0, -2, -3, 1, 0, 0, 0, -1, 2, 4, 1, -4],
    [0, 1, 0, 2, 0, 2, -2, -2, 0, 2, 4, 0, -4, 1, -3],
    [0, 0, 0, 0, 0, 0, 0, 1, 0, 2, 0, -2, -3, 5, -1],
    [1, 1, 0, 1, 2, 3, -2, -1, 0, 1, 2, -2, -4, -1, 2],
    [1, 0, -1, -1, 1, -1, 1, 0, -1, -1, -1, 5, -1, -4, 5],
    [0, 1, 0, 2, -1, 0, -3, -2, 0, 2, 1, -4, 3, 4, -3],
    [1, 1, 0, 2, -1, 1, -3, -4, -1, -1, 1, 0, 2, -2, 0],
    [-1, -1, -1, -2, 1, 0, 2, 1, 1, 0, 0, 1, 2, -3, 0],
    [0, 0, 0, 0, 0, 0, 0, -1, -1, 0, 1, 4, -2, -2, -1],
    [0, 0, 0, 1, 0, 1, -1, -1, 0, 0, 1, -3, 5, -2, -3]
])
S
```

```
S.inverse()*A*S
```