Learning Opportunity 1-3

AIMS-ZA Advanced Linear Algebra with Sage

Robert A. Beezer

January 31, 2018

1-3-1 This exercise will employ Householder matrices and prepare you for our discussion about QR decompositions. Copy the `house()` function from the day's Sage demonstration for use here.

1. Build a random 10×10 matrix of rational numbers and then use the `.change_ring()` method to convert the entries to algebraic integers. Call the result A.
2. Extract the entries of the fourth column of the matrix from rows five though ten and call this vector u. (Remember, Sage/Python starts indexing at zero and mathematicians start counting from one.) Build the Householder matrix for u and call it W.
3. Make a new 10×10 matrix, Y, with Sage's `matrix.block_diagonal()` constructor using an identity matrix for the first block and your Householder matrix W for the second block.
4. Verify that Y is Hermitian and unitary.
5. Compute the matrix product YA and provide careful, explicit, readable commentary on the result. (In other words, create a Markdown cell in your Jupyter notebook and use LaTeX syntax, etc. to craft a well-written paragraph.)

Full marks for a correct Y, with checks, and a well-written commentary about the computation YA.

1-3-2 This exercise will help you get acquainted with sets of orthonormal vectors, their construction and properties. You will also use various Sage commands to manipulate parts of matrices.

1. Build a 5×5 random nonsingular matrix of integers, but over the field of rational numbers, with a command like

   ```
   random_matrix(QQ, 5, algorithm='unimodular', upper_bound=10)
   ```

   and call this matrix D. Run this enough times that you get a matrix without too many zeros.
2. Run Sage's Gram-Schmidt command on the *columns* of D and extract a list of vectors, which are an *orthogonal* (not necessarily orthonormal) set of vectors that span $\mathbf{C}^5$. You will definitely have to read, study, and understand Sage's documentation on the `.gram_schmidt()` method. This is part of the exercise, and the documentation is reasonably clear, though you may need to read carefully. You may find the `.transpose()`, `.rows()` and `.columns()` matrix methods helpful.
3. Construct a matrix P whose columns are your orthogonal set. Compute P^tP, which will be a diagonal matrix whose entries are the lengths squared of the columns of P. Scale each column of P to be a unit vector (length/norm 1) and call the new matrix Q. Important: wrap your square root in `QQbar()` to express the square roots as algebraic numbers (not symbolic expressions, which is Sage's default). Check Q^tQ. Discuss what you observe with a well-written commentary.
4. Now the columns of Q are an orthonormal basis. Create a random vector of integers with a command like `v = random_vector(ZZ, 5)`. Read, study amd understand Theorem COB in FCLA, which says that we should be able to write v as a linear combination of the columns of Q. Compute a list of the scalars for this linear combination in two ways, each time with a single line of Sage code. First, use a linear system of equations. Second, use a list comprehension and facts from Theorem COB. Include a verification that your scalars do the job they should do.

Full marks for a correct Q matrix and well-written commentary, along with both versions of the linear combination code with a check for each.