Lab 7: Principal Components Analysis (PCA)

Due TBD

Goals:

1) To implement Principal Components Analysis (PCA) in MATLAB;
2) To gain experience manipulating remotely-sensed images of the Moon.

Introduction:

In this lab, you will be working with multispectral imagery of the Moon taken by the Galileo spacecraft (http://galileo.jpl.nasa.gov). The data you have are from the Solid State Imager (SSI) that was on board Galileo and are ultraviolet-visible images of Imbrium (http://en.wikipedia.org/wiki/Mare_Imbrium), one of the largest mare-filled impact basins on the Moon.

Principal Components Analysis (PCA)

Multispectral datasets comprise a set of variables (the spectral bands), which are usually correlated to some extent. That is, variations in the DN in one band may be mirrored by similar variations in another band (e.g., when the DN of a pixel in Band 1 is high, it is also high in Band 3). This correlation has two adverse effects on data processing. First, there may be redundancy in the data that causes us to work with more data than we actually need to in order to capture the full information content of the imagery. This visually manifests itself in false-color RGB imagery that tend to have subdued colors. Second, the correlation will convolve useful information with noise.

PCA allows the analyst to determine the "dimensionality" of multispectral data, that is, how much "information" it contains. PCA can then be used to separate the most useful information from less useful (redundant) information and both from noise. The benefit is more efficient and more informative representation of image data. Essentially, PCA puts the input image bands through a linear transformation (statistically-based on the covariance matrix) which reorganizes their variance to create a new set of output image bands. These output image bands are called the principal component (PC) bands. By doing this, you transform a set of correlated variables (input bands) into a new set of uncorrelated variables (PC bands). These PC bands can then be analyzed and subset to perform a variety of tasks including image compression and filtering, as well as to understand the source of various components of spectral signatures in the original data.

Since we are talking about multispectral images with n bands, we can think of PCA as a coordinate transformation done in n-dimensional space. Every pixel in the input image (and also in the resulting PC image) can be thought of as an n-dimensional vector. These vectors are lists of n numbers, each of which is the BV at a particular pixel location in each of the n bands. The transformation is done by finding a set of orthogonal axes (eigenvectors) that have their origin at the image data mean and that are rotated so the image data variance is maximized (see lecture notes and reading). The n-dimensional eigenvectors define the orientation of the principal component axes in n-dimensional space.

There is one eigenvector for each PC band, and each eigenvector contains one element for each input band (a total of *n*). For example, PCA of a 14-band input image would produce 14 PC bands and 14 corresponding eigenvalues and 14 associated eigenvectors that each have 14 elements (where the first element of the eigenvector corresponds to input band 1, the second element to input band 2, etc.). Mathematically, the eigenvector elements are the weighting coefficients in the linear transformation from input image bands to PC bands such that the value of a particular eigenvector element is proportional to the corresponding input band's contribution to the corresponding PC band. The eigenvector that corresponds to the highest eigenvalue tells us the dimension (axis) that accounts for the maximum amount of variance in the image data.

The resulting PC bands are linear combinations of the input (spectral) bands and will have segregated different components of the original image data. Each successive principle component accounts for a progressively smaller proportion of the variation in the original data. For example, the first 3 or so PC bands (ordered by decreasing eigenvalue) will usually contain >90% of the "information". In addition, each principal component is dominated by a particular aspect of the data. For example PC band 1 (the one with the highest eigenvalue) will be dominated by information with the highest variance among the original bands. Higher number PC bands (those smaller eigenvalues) will information with less variance and that is more redundant (e.g. found in all the bands), for example noise. Analysis of a factor-loading matrix computed from the eigenvectors, eigenvalues, and spectral band variances, can be used to determine which of the original spectral bands contributes (and how much) to each PC band.

Instructions:

You will work with the 6-band, ENVI-format stack of Galileo SSI data provided to you on `//geotwo`. The ENVI header has information on its spectral and spatial resolution and spatial extent. **Write a MATLAB program to perform PCA and test it on this image dataset. Your code should also work on any image, so be sure to make your code as generic as you can to accommodate other image data.** Use your covariance matrix program from the second lab as the starting point, and follow the procedure outlined in the notes. Your MATLAB program should: (a) calculate the covariance (and correlation) matrix (*use your functions from Lab 2*); (b) calculate the eigenvectors for the covariance matrix (*use MATLAB's functions*); (c) calculate the associated eigenvalues (*use MATLAB's functions*); and (d) perform the PC transformation (*write your own function here – this is the most important part of the lab*). It should also: (e) display the resulting PC images; (f) compute the percentage of the total variance explained by each of the principal components; and (g) compute the factor loading matrix (see notes and readings). If your covariance/correlation matrix program works, you've already done a lot of the hard part! For the linear algebra operations, the MATLAB function `eig` will do what you need. Below are a set of questions to answer based on the results of your PCA program:

1) In ENVI (or MATLAB if you prefer), look at your *original* image bands (the input data). Which look most alike? Which are most dissimilar? Experiment

with making false color (i.e. RGB) displays in ENVI (or MATLAB if you prefer) using the *original* bands. What do they look like? What might you learn about the surface composition of the Moon from these?

2) Perform a PCA transformation on your image data using your MATLAB program.
3) Consider the degree of intercorrelation (data redundancy) in your *original* data by examining the *covariance and correlation matrices*, *eigenvalues and eigenvectors*, and the *factor loading matrix* (*don't look at the images yet*) generated by your MATLAB program (look back in your notes on statistics). What can you say about your data based on the statistics, e.g.: How much variance is accounted for in each of your *PC* bands? What *pair* of *original* image bands are the *most* correlated? Which *pair* of *original* image bands are the *least* correlated? Which of the *original* image bands contribute to which of your *PC* bands? Which of the *original* image bands carries the most information? **Be sure to justify each of your answers here with statistics!**
4) Look at and describe the *most correlated PC* band (the one with the largest eigenvalue). Which of your *original* image bands does it most resemble? What might this mean? What does the factor-loading matrix tell you in this respect (refer to your answer in 3)? Geologically (or otherwise) what do you think this image represents? Also look at and describe *each* of the other *PC* bands, including the *least correlated PC* band (the one with the smallest eigenvalue). What sort of information is contained in each one (geologically or otherwise)? Again, how does the factor loading matrix help you here?
5) If, based on the PC results, you had to choose three of the original image bands to display as an RGB image, which would they be and why?

What to Turn In and How:

Turn in your MATLAB code (.m file) and also include a *short* (1 page *maximum*, not including figures/tables) report answering the questions posed above. Illustrate it with representative images, tables of statistics, or anything else you *need* (no superfluous stuff please!) to answer the questions. For the report, submit a single PDF file that includes both the text and figures. Do not turn in any other loose files. In your `.m` files, be sure to put your name, date, etc. in a comment at the top. Also be sure to adequately comment your code so that I know what your programs do and how they work. Turn in your files via a folder that you should place in the dropbox on `//yowa`.