MRI Practical – MRI/Matlab Focus
Non-Ionising Functional & Tissue Imaging

Dr. Neal Bangerter, February 2021

Part 4: Measuring Signal-to-Noise Ratio (SNR) and Contrast-to-Noise Ratio (CNR) from Images

A common way of determining the signal-to-noise ratio (SNR) of a tissue in a medical image is to measure the average (or mean) signal across a region of the tissue in question, and then divide by the standard deviation of a region in the background of the image (where there is presumably only noise). The SNR gives you a measure of how well the tissue in question can be distinguished from the background noise present.

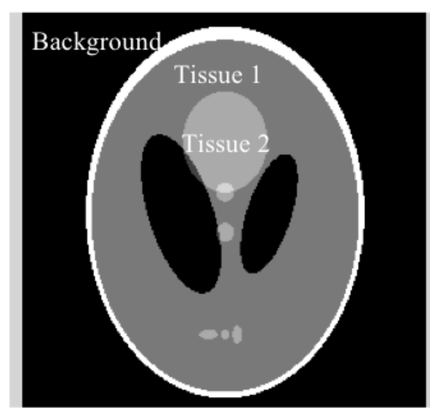
Sometimes we are interested in how well we can tell two different tissues apart in a medical image. In this case, the so-called contrast-to-noise ratio (CNR) is a useful measure. The contrast-to-noise ratio is commonly defined as the **difference** between the mean signals of two different tissues, divided by the standard deviation of a region in the background of the image. The larger the separation of the average signal between to tissues relative to the background noise, the easier they will be to distinguish from each other.

The "Shepp-Logan phantom" is a computer-generated fake medical image that is often used in medical imaging to test various image processing or reconstruction schemes. In this problem, we will add noise of various types to the Shepp-Logan phantom image and then measure SNR and CNR for various tissues in the phantom.

(1) Load the Shepp-Logan phantom (cast to double) and add Poisson distributed noise with parameter a=80 to the image. (**NOTE: Please review and refer to the January 31st lecture note supplement on Random Variables.** A Poisson random variable with a=80 has both mean and variance of 80). You can generate a 256x256 array of Poisson distributed random noise with a=80 with the command:

```
n1 = poissrnd(80, 256, 256);
```

Verify that your noise image n1 does indeed have a mean of about 80 and a variance of about 80. The 'mean' and 'var' commands in MATLAB will be helpful here. You can also quickly "flatten" a 2D array into a 1D vector by using the notation n1(:) in Matlab.



(2) Compute the SNR of Tissue 1 (as shown in the image to the right) by measuring the average signal value across a region in Tissue 1 and dividing by the standard deviation of the signal in a region of the background. **NOTE: It is easiest to simply choose a rectangular region in each tissue across which to measure average values and standard deviation. Make sure you measure a large enough region in both the tissue and the background to get good statistics. For example, make sure you use at least a 15x15 pixel area (225 pixels) in the background for the standard deviation. The matlab command 'std' will give a standard deviation.**

(3) Compute the CNR between Tissue 1 and Tissue 2 in your image, and provide the measured CNR with your homework. As stated previously, it is easiest to simply choose small (e.g., 15x15 pixel) rectangular regions in each tissue across which to compute average signal levels.

Part 5: Modeling Bivariate Gaussian Noise in MRI

In the previous problem, we dealt with a real- and positive-valued image, and added Poisson-distributed noise. This is a good noise model for many x-ray systems, but **is not** an accurate model for magnetic resonance imaging (MRI). In MRI, we acquire **complex-valued** data in the Fourier domain, and (to a very good approximation) we have zero-mean Gaussian noise on both the real and imaginary channels in the Fourier domain. You will model this "bivariate Gaussian noise" case below:

(1) Load the Shepp-Logan phantom and cast to a double.

(2) Transform the Shepp-Logan phantom into the Fourier domain.

(3) Our Fourier-domain image is complex-valued. In MRI, we find Gaussian noise on both the real and imaginary channels in the Fourier domain. Create zero-mean Gaussian noise with standard deviation of 5000 on both the real and imaginary channels with the following commands:

```
n3 = normrnd(0, 5000, 256, 256) + j*normrnd(0, 5000, 256, 256);
```

(4) Add this noise to the Fourier domain data of the Shepp-Logan phantom that you created in step (2).

(5) Inverse Fourier transform the noisy Fourier-domain data back into image space, and display the absolute value of the image.

(6) Repeat the SNR and CNR measurements from the previous problem on the absolute value of the new noisy image.