Quantitative and Functional Imaging

BME 4420/7450

Project #3

**Segmentation and volumetrics**

The goal of this project is to measure the total volume of Multiple Sclerosis (MS) lesions using multispectral data. As in the previous projects, you are free to get your results in some other way—these procedures are just one (not necessarily optimal) method. Matlab commands are given in *italics* for easy reference. Use *help <command>* (for example, *help ind2sub*) or the Matlab Help pages for more details on any Matlab function.

1. Load the data file proj3Data.mat into your Matlab workspace. There are **three** arrays in the file:

pdw\_m A matrix (256 x 256) of pixel values for a proton density (PD) weighted image.

t2w\_m A matrix (256 x 256) of pixel values for a T2 weighted image.

t1w\_m A matrix (256 x 256) of pixel values for a T1 weighted image.

These are three images of the same slice, with three different (dominant) sources of contrast. Display the **three images** in one figure.

1. Create a Matlab program to prompt the user to identify (click on) several pixels in MS lesions. These are called “training points.” By defining a few of these by hand, you will give your program enough information (in principle) to identify the other lesion pixels in the image. Find the coordinates of the training points:

*nTrain = 6;*

*figure*

*imagesc(t2w\_m)*

*colormap(gray)*

*axis image*

*disp(['Click on’, num2str(nTrain), ‘ lesion points'])*

*[x\_v, y\_v] = ginput(nTrain); % get at multiple locations to gain some robustness*

*row\_v = round(y\_v);*

*col\_v = round(x\_v);*

The *ginput(n)* command returns the x and y coordinates of ***n* mouse clicks** in the current figure. Draw a symbol (e.g., a red dot) on the T2-w image at each training point. Find **the PD and T2-weighted image intensities for the training points**:

*index\_v = sub2ind(size(t2w\_m), row\_v, col\_v);*

*t2wl\_v = t2w\_m(index\_v); % T2 weighted intensity for lesion points.*

*pdwl\_v = pdw\_m(index\_v); % PD weighted intensity for lesion points.*

Matlab can index matrix elements in two ways: **by (row, column) coordinates** and by **a single index**. The single index is the number of the element starting from the top left corner (row = 1, column = 1) and counting down the columns (after the first column, counting continues at the top of the next column). For example, for our 256 x 256 matrices, element 257 is at the top of the second column, element 513 is at the top of the third column, and so on. The *ind2sub* command calculates the (row, column) coordinates corresponding to a given index (for a given matrix size), and the *sub2ind* command calculates the index for given (row, column) coordinates.

Next find the **mean PD and T2-weighted image intensities** for the training points:

*t2wl = mean(t2wl\_v);*

*pdwl = mean(pdwl\_v);*

Repeat for the other **important tissue classes**, using the T2-weighted image to define the training points. Place symbols on the image for each training point (a green ‘+’ for white matter, a blue ‘x’ for gray matter, and a white ‘\*’ for CSF). Calculate the **pixel** and **mean** intensities for white matter:

t2ww\_v, t2ww (T2-weighted intensity)

pdww\_v, pdww (PD-weighted intensity),

gray matter:

t2wg\_v, t2wg (T2-weighted intensity)

pdwg\_v, pdwg (PD-weighted intensity),

and cerebral spinal fluid:

t2wc\_v, t2wc (T2-weighted intensity)

pdwc\_v, pdwc (PD-weighted intensity).

1. Plot the training point intensities in the **T2 versus PD** “**feature space**:”

*figure*

*plot(pdwl\_v, t2wl\_v, 'r.', pdww\_v, t2ww\_v, 'g+', ...*

*pdwg\_v, t2wg\_v, 'bx', pdwc\_v, t2wc\_v, 'k\*')*

*xlabel('PD-w intensity')*

*ylabel('T2-w intensity')*

The *plot* command displays a scatter plot of the lesion, white matter, gray matter, and CSF training points in feature space. The lesion points are displayed as **red dots** (‘r.’), the white matter as green plus signs (‘g+’), the gray matter as blue x’s (‘bx’), and the CSF as black asterisks (‘k\*’).

1. Use the **PD-weighted image** to form **a mask of the head, and find the location and image intensities for each pixel:**

*pdMax = max(pdw\_m(:));*

*mask\_m = (pdw\_m > 0.1\*pdMax);*

*index2\_v = find(mask\_m(:));*

*t2wHead\_v = t2w\_m(index2\_v);*

*pdwHead\_v = pdw\_m(index2\_v);*

*nHeadPixels = length(index2\_v);*

The colon operator, :, reshapes a **multidimensional array into a long 1D array** (a column vector). Hence, mask\_m(:) is a column vector with 256x256 = 64k elements: the first column of mask\_m comes first in the 1D array, the second column of mask\_m comes next, and so on. The **vector index2\_v** holds the indices of the non-zero elements of mask\_m(:).

1. For each pixel in the image, find the tissue type that is closest in the feature space. To save time, do the calculation only for pixels in the head (i.e., where the mask equals 1):

*lesionMask\_m = zeros(256, 256);*

*wmMask\_m = zeros(256, 256);*

*gmMask\_m = zeros(256, 256);*

*csfMask\_m = zeros(256, 256);*

*for pixel = 1:nHeadPixels*

*% Find intensities for current pixel:*

*t2w = t2wHead\_v(pixel);*

*pdw = pdwHead\_v(pixel);*

*% Insert code here to find the distance in feature space*

*% between the current point at coordinates (pdw, t2w)*

*% and the mean position of the training points for each*

*% of the four pixel types (lesion, white matter, gray matter,*

*% and cerebral spinal fluid).*

*% Insert code here to find which pixel type the current*

*% point is closest to.*

*Compare*

*% Make the (row, col) element of the corresponding tissue*

*% mask (lesionMask\_m, etc) equal to 1. Note that the*

*% current pixel’s position in the image is given by*

*% [row, col] = ind2sub(size(t2w\_m), index2\_v(pixel));*

*end*

1. Display the segmentation map:

*figure*

*image(cat(3, lesionMask\_m, wmMask\_m, gmMask\_m))*

*axis image*

*axis off*

If the *image* command is passed a 3D array (with elements in the range [0, 1]), it will display a color image. The first page of the array defines the red component of the image, the second page the green component, and the third page the blue component. Hence, lesion pixels are displayed in red, white matter is displayed in green, and gray matter is in blue (CSF is black).

1. To exclude the extracranial (outside the skull) tissues (mostly fat) from the lesion class, draw a polygon in the skull, enclosing the brain. The skull, which does not contain much free water, appears dark on MRI images. Use the *roipoly* function to calculate a mask that is zero outside the skull:

*disp('Define a polygon enclosing brain, excluding extracranial fat')*

*skullMask\_m = roipoly(pdw\_m / pdMax));*

*% Zero all pixels outside the skull:*

*lesionMask\_m = skullMask\_m .\* lesionMask\_m;*

*wmMask\_m = skullMask\_m .\* wmMask\_m;*

*gmMask\_m = skullMask\_m .\* gmMask\_m;*

*% Display improved segmentation map:*

*figure*

*image(cat(3, lesionMask\_m, wmMask\_m, gmMask\_m))*

*axis image*

*axis off*

The *roipoly* function allows the user to define a polygon on an image. Each mouse click defines a new vertex of the polygon--a double-click closes the polygon (by drawing a line from the double-clicked point to the first point). Right-click on the polygon and select the ‘Create Mask’ option so *roipoly* returns a binary mask.

1. Find the total number of lesion pixels by **adding all the 1’s** in the lesionMask\_m matrix.
2. Find the T1 weighted image intensity for each training point. Plot the **training point positions in the 3D feature space of image intensities** (PD, T2, T1). Note whether T1 weighting provides information not available from T2 weighting.

***With T1 weighting, we get extra information for distinguishing between white (green) and grey matter where white matter has higher T1 intensity than grey.***

1. Repeat the pixel cla**ssification step using T1 weighted image intensity in addition to the T2 and PD information**. Display the resulting (**three parameter**) segmentation map. **Does the use of T1 weighted image intensity improve the accuracy of your segmentation?**

# Questions

1. How accurate is your final segmentation? If there **are regions where the tissue classification seems wrong**, indicate these on your segmentation map (you can just draw a circle around them, for example).
2. Can you think of any ways to make the segmentation more accurate? What would you change in your algorithm or image acquisition? Can you think of another way to exclude the extracranial fat?
3. What is the total number of **lesion voxels?**
4. According to your 3D feature space plot, does T1 weighting provide information not available from T2 weighting? Does it significantly improve the segmentation map?

# Assignment

Create a document that includes

1. The three images of different contrast (PD, T1, and T2). Please label your images
2. **T2-weighted image showing the locations of all training points (step 2).**
3. The plot(s) of training point locations in feature space (steps 3 and 9). Please provide a **legend** labeling the tissue types.
4. The segmentation map(s) (steps 6 and 10).
5. Your answers to the questions above.
6. Your Matlab code.

Save your report as a PDF file, name it “Project3…” (adding your name), and submit it on Brightspace by Tuesday, Oct 24. Each group can submit one report—just make sure all group members are named on the report.