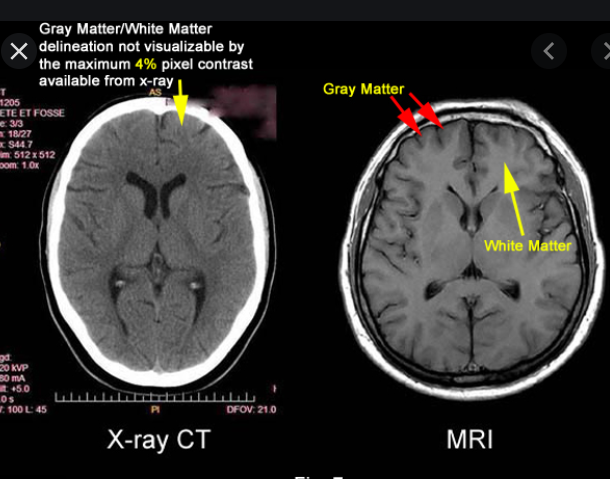
# Medical Image Computing

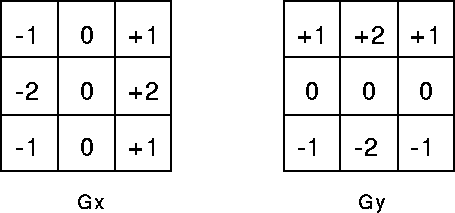
# Question 1

## Part a

Left looks like CT scan and right looks like MRI (don’t know if this is examinable)



## Part b



Applying these to the image we get (I think):

|  |  |  |
| --- | --- | --- |
| 4 | 0 | -4 |
| -4 | -4 | -4 |

For the horizontal Sobel filter, and

|  |  |  |
| --- | --- | --- |
| -8 | 0 | 8 |
| -8 | -4 | -8 |

for the vertical filter.

## Part c

The window level (window center), is the midpoint of the range of the CT numbers displayed. When the window level is decreased the CT image will be brighter and vice versa.

Midpoint of [-200,700] = 250 (??)ni

# Question 2

## Part a

The partial volume effect can be defined as the loss of apparent activity in small objects or regions because of the limited resolution of the imaging system.

From <https://en.wikipedia.org/wiki/Partial_volume_(imaging>).

In essence, this is where the pixels/voxels don’t exactly match up to reality because we only have a quantised version.

Other challenges include:

* **Anisotropic Resolution:** where the resolution in one plan is different to the resolution in another.
* **Imaging Artifacts:** such as streak artifacts caused by metal artifacts.
* **Limited Contrast:** where boundaries between objects may not be visible.
* **Morphological Variability:** makes it hard to incorporate meaningful prior information or shape models.

I think leakage means when pixels that do not belong to the class of interest are segmented out due to limited contrast of neighbouring segmented regions.

## Part b

I don’t think he went through the details of an algorithm for region growing, but the gist is that, starting from a user-supplied seed point you grow the region according to an intensity threshold.

Start at specified seed point, set class label to X

for each neighbour of current pixel make recursive call to do the following:

if abs(current\_pixel\_intensity - seedpoint\_pixel\_intensity) < threshold:

Set current pixel class label to X

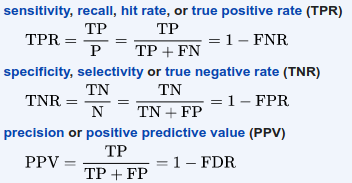
Call recursively again on neighbouring pixels

In multi-atlas label propagation, we start by running a **registration algorithm** which performs a spatial transformation to register each labelled scan with each unlabeled scan. Then, we can perform **label propagation** to map labels to the unlabeled scans. Since we have a *set* of labels for each new scan, we need to perform **label fusion** to reconcile them. This can be done using simple majority voting, or probabilistically.

## Part c

* **True positives:** 10
* **False positives:** 2
* **True negatives:** 18
* **False negatives:** 6

Sanity check: numbers sum to 36.

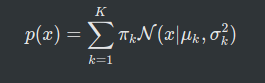


* **Precision** 0.83
* **Recall** 0.63
* **Specificity** 0.9
* **Dice** 0.72

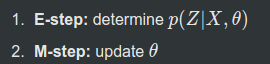
Specificity (or true negative rate) measures how well we are doing on pixels that are part of the background. This is not that interesting because we can increase this by simply having an image with more background pixels.

# Question 3

## Part a



In EM, latent variables determine the assignment of observations to mixture components.



I have no idea what Variational Bayesian estimation is. Me neither :(

I think they want to hear something about hyperpriors here so we don’t have to choose our number of clusters, but instead they are inferred.

The EM only relies on the observed data to estimate the parameters and VBE-GMM incorporates the hyperpriors about the parameters.

## Part b

More training data will likely help in scenario B. Low training but high test error indicates that the model is unable to generalise well to unseen data. More training data will likely help alleviate this.

(Low train, high test = overfitting, in which case more data prevents the model from overfitting as much)

(low train, low test = underfitting, more data won’t help the model as it already cannot fit the less data)

In case A, we could also try and increase the model complexity. In case B, we could try the opposite, by reducing model complexity or we could try adding regularization (e.g. Lp, dropout if NN...).

## Part c

The final prediction would be class 2. Perhaps this could be considered unusual since all three classifiers have relatively low probability? This suggests that the sample is (according to the model) unlikely to belong to any of them.

In a one-vs-one strategy, each classifier would distinguish between two classes. We would need K(K-1)/2 classifiers. The final prediction can be decided using majority voting.

We cannot use the squared error cost function because the fact that the hypothesis function is non-linear means the cost function will not be convex. This means the optimization landscape may have many local optima, making it difficult to find the best solution.