CS188 Project Report
Beiqi Guan
804160726

Machine Learning Technique in Co-registering MRI Images

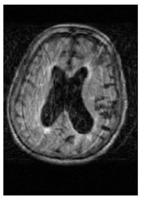
Objective

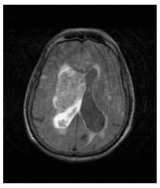
In this project we aim to implement a method to perform automatic co-registration of two multi model MRI images, which belong to the same patient but acquired in different days. The main method I attempted was Multi-linear Regression, and the result turns out to be satisfactory.

Procedure

We want the software to be able to perform co-registration automatically, but how do we do that? The preliminary approach I took was to use multi-linear regression model, and then I use the coefficients calculated to generate estimated result.

Preliminary Training and Testing using Multilinear Regression





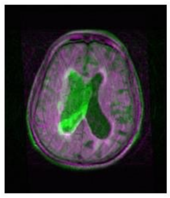


Figure 1, 2 and 3: figure 1 is taken from flair1 in 1.mat, figure 2 from flair3, while the third is the co-registration result of two. Note that the size of figure 1 and figure 2 differs, so we need to apply affine transformation to figure 1 in order to co-register correctly.

We want to train the model to know what is a good match, so we match the images first on our own. Check figure 1, which is a MRI image of patient's brain at day 1, and figure 2, the MRI image of patient's brain at day 3. We manually pick control points in the two images, and create affine transformation matrix based on that. Notice that in figure 3 we are able to perfectly co-register these two images.

After successful co-registration, we can extract training sets from these two images: we pick points inside brain and create patch windows. At this point, we know a pair of patch windows at the same location in two images should be seen as a good match. We want to create good matches and bad matches to train the model: patch windows generated at the same point in two images should be seen as a good match, while patch windows that do not sit at the same center position should be treated as a bad match.

We need two values to train the model: X and Y. Now that we have the good match and the bad match, we will label good matches with value 1 in Y, and bad match with value 0. X will be the pixel values in the patch windows of image 1 and image 2: we vectorize the value so it will have the dimension n*1. Every entry of X should match a pair of patch windows selected, and should contain the pixel values for these two patch windows. For instance, if or patch window size is 5*5, X entry for a pair of patch windows selected should have the dimension 50*1.

Training and Validation

We want to have enough Xs and Ys to generate meaningful coefficients that give us a correct classification of points provided X. For one patient, 50 sample patch windows were picked, with window size 9*9. If we run regression based on these data, we will get a 192*1 vector as the coefficient between Y and Xs. In Run 1, we followed the above procedure for data in "1.mat".

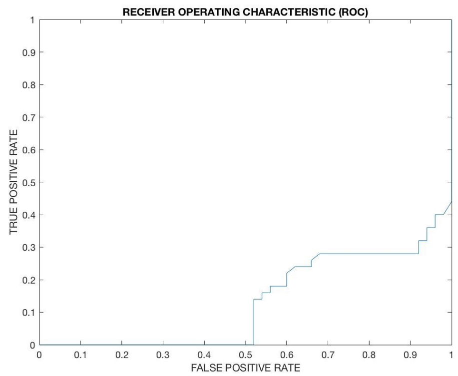


Figure 4: ROC graph of Run I (AUC = 0.87)

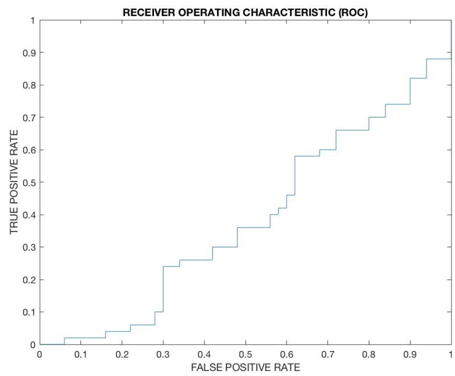


Figure 5: ROC graph of RUN II (AUC = 0.6176)

In order to measure the accuracy of the model, AUC score is needed. Using coefficient obtained and the original X, Y', the estimated score for every patch was obtained. Then, AUC score was calculated using Y and Y', and was used to produce the ROC graph in Run 1 as shown in figure 4. As we can see, AUC score is 0.87, which indicates a quite fitting relationship. Such high score shows potential for the multilinear regression model.

However, in order to get a correct measure of the performance of the model, we need to generate coefficient b using multiple patients' data, and test it on a patient's data that was not used in the training stage. This approach can be seen as a preliminary version of cross validation.

Thus, we started Run II, where X and Ys were collected from 2.mat to 15.mat, and were then used to generate coefficient b. We then used b on data in 1.mat, giving a Y estimated for patch windows in 1.mat. The result is shown in figure 5.

As we can tell, the AUC gives a much lower percentage compare to Run I, but still seems reliable enough to give accurate estimate for Y.

Dilemmas

Even though the validation tells that the multi-linear regression method is working, it does not apply to all the cases. If we test coefficient b on newly picked random patch windows, it does not give a satisfactory AUC score. The reason causing this might be that the pixel value across samples were not correctly normalized. Future work can be done to improve performance in this case and prevent this from happening.

Also, obvious decrease in AUC score if we obtain coefficient b using multiple patients' data. This is natural but it will be better if we can keep AUC score around 70%.

Future Work

The next step will be using the coefficient b to find possibly good image transformation matrix, which can help us co-register two MRI images for one patient. Given range of angle and scale in affine transformation, we can perform a through evaluation on each of them: using existing b and randomly picked patch windows, we expect to find the transformation that gives us highest Y, which means that all patch windows co-register.