CIS 537 Segmentation Grand Challenge – Part 1

Due Monday 3pm April 27, 2015

1. Introduction

In this assignment, you will have the chance to investigate and compare three different segmentation algorithms that all have a Markov Random Field (MRF) formulation. The first two algorithms - Iterative Conditional Mode and Graph Cut - were discussed in class on Monday March 30. The third algorithm, the Random Walker segmentation, was briefly discussed in class on April 1 and is fully described in two papers included with the assignment.

In the first part of the challenge you are asked to produce an improved segmentation of the hippocampus in the data sets you used for the Registration Challenge. This will be an multi-atlas-based segmentation using a provided affine registration to align a template of the hippocampus to each data set and then refining the segmentation with the MRF based algorithms. The output should be images at the resolution of the original images but with a label of one for the hippocampus and zero for anything not the hippocampus.

The second part of the challenge will involve applying the 3 MRF segmentation algorithms to a different data set and a different application. For the second part you will not be using an atlas-based initialization. Instead, you be provided with samples of the image that have been hand-labeled and then the algorithms will need to label the rest of the image. This data set and additional instructions will be provided later. Both parts of the Challenge are due April 27 but you should plan to have the first part mostly done by early in the week of April 20 so that you will have time to work on the second part.

2. Multi-Atlas initialization

The hippocampus is a relatively small structure relative to the size of the brain and consequently it is a hard problem to get an accurate registration for all data sets with a single image as an atlas. Consequently, you should implement a majority voting multi-atlas scheme to initialize each of the MRF segmentation algorithms. When testing on a specific image you can use any or all of the other images as atlases. For the challenge on unseen data you should implement a specific strategy that can use all of the training data. In the report you must explain how you did an evaluation on the training data and explain what you submit for the challenge.

3. Segmentation

You will need implementations for all three segmentation algorithms.

3.1 ICM

Matlab code for the Iterative Conditional Mode was provided in class and is available in canvas in **mrfcode/classify.m**. You should make a separate function for the ICM algorithm. You can use the core algorithm as is or modify it as you see fit. In your report and in the matlab code explain any modifications.

3.2 Provided MATLAB Routines for the Graph Cut algorithm

The assignment includes an implementation of the Boykov and Kolmogorov algorithm for Graph Cuts in MATLAB. In MATLAB, you will need to enter the directory **library/maxflow** and run the command **make** to compile the MEX code in the library. You might see a lot of warnings from the compiler, but there should not be any errors. Then run commands **test1** and **test2** to make sure everything compiled correctly. Finally, add the directory **library/maxflow** to your MATLAB path (using **addpath**).

The implementation includes two useful routines:

> **edges4connected**
>
> Can be used to encode the n-links in an image as a sparse adjacency matrix, which can be passed in as the first input to **maxflow**. Note that this function assigns weight 1 to every n-link.
>
> **maxflow**
>
> Computes the minimum cut for a graph. Inputs are sparse matrices encoding the weights of the n-links and t- links. Note that the inputs must be sparse matrices. You can use the MATLAB command **sparse** to create a sparse matrix from a dense matrix.

The test routines **test1.m** and **test2.m** illustrate the use of these commands.

Other Allowed MATLAB Routines

The paper Boykov and Funka-Lea 2006 describes how the graph for the Graph Cuts algorithm is built of two kinds of ledges (also called links): n-links (which encode the regularization term in the energy function) and t-links (which encode the data fidelity term). To compute the data term (weights of the t-links in your graph), you will need to use an algorithm that computes for each voxel the probability that it belongs to each of the six tissue classes, based on the examples of each class provided by the user in the form of the squiggles. This can be done by estimating a probability distribution for the image intensities such as a Gaussian distribution or by defining a classification algorithm. There are many classification algorithms available (K nearest neighbors, random forests, so on) and you are free to choose whichever one suits you best and to use to corresponding MATLAB routines. If you are not familiar with classification and you would like to try a classification algorithm, I would suggest using K nearest neighbors. You can find out more about K nearest neighbors (different from the K-means algorithm) in Wikipedia (http://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm). MATLAB provides an implementation (**ClassifierKNN.fit** and **predict** functions, which you are allowed to use).

3.3 Random Walker segmentation

For the Random Walker segmentation algorithm you will need to read the technical papers and make your own implementation.
The Random Walker segmentation algorithm is described in:

[1] "Random Walks for Image Segmentation" Leo Grady, IEEE Trans PAMI 28(11) 2006. A second paper explains how to incorporate a data fidelity term into the formulation of the first paper which only considered a constraint based on the relative intensity of neighboring pixels (2-cliques).

[2] "Multilabel Random Walker Image Segmentation Using Prior Models", Leo Grady, Proc. CVPR Vol 1, pp 763-770, 2005.

The Random Walker segmentation algorithm is solved by solving a large sparse system of linear equations – equation (10) in [1]. Consequently it is relatively easy to implement in matlab. There are several online solutions (including one from Leo Grady). You should do your **own implementation!** An important part of understanding the algorithm is understanding how to set up the specific system of linear equations to get a useful result.

4 Data

The full dataset has been divided into training and testing subsets. For the images in the training set, manual segmentations are provided, which allows you to test your algorithms. The segmentations for the testing set, which are not provided, will be used to test your algorithm on unseen data.

The **data/mri-hippocampus** data is 2D sagittal samples of the data you worked with in the Registration Challenge.

5 Evaluating the segmentation results

You will need to build tools to measure the quality of your segmentation results. Most measures of segmentation quality fall into two categories: overlap measures or measures of the distance between boundaries.

- As an overlap measure you should implement the **Dice coefficient**. The Dice coefficient is defined for two segmentations A and B.

  $$\text{Dice} = 2 * |\text{intersection}(A,B)| / (|A| + |B|)$$

  where $|X|$ is the size of the region X in pixels.

- For a measure of the distance between boundaries you should implement the following measures

  Average distance: this is average of the minimum distances between each point on the boundary of the ground truth region and the computed segmentation.

  Maximum distance (Hausdorff distance): this is the maximum of the minimum distances between each point on the boundary of the ground truth region and the computed segmentation.

You will likely want to make use of the MATLAB function **bwdist** to help you in computing the average distance and maximum distance.

You should report all 3 values (Dice, Average distance, Maximum distance) for each application. However, the Grand Challenge results will be determined base on the best Dice overlap measure.

6 What you need to do

6.1 Basic assignment

The labeling from the atlas through the registration alignment will serve as the input to a segmentation of the hippocampus in the target image. You will then use each of the three MRF segmentation algorithms to segment the image into either hippocampus (label 1) or not-hippocampus (label 0).

In your report you should explain how you chose your multi-atlas and how the labeling from the atlas serves as input to start each of the segmentation algorithms.

6.2 Possible extra work:

For the Graph Cut and Random Walker algorithm you may want to consider several options for how to define the edge weights and hence the data fidelity term and the regularization term. The regularization term is the one that looks at values of pairs of pixels.

You may also want to investigate how a neighborhoods are defined using either 4-connected or 6-connected neighbors of a pixel.

For some ideas regarding the two extensions above it may be helpful to look at:

[3] "Weights and topology: a study of the effects of graph construction on 3D image segmentation" L. Grady and MP Jolly, MICCAI LNCS 5241 pp 153-161, 2008.

Each of the 3 MRF algorithms is formulated to take a fixed data fidelity term. Consider including one (or more) of the 3 algorithms into an iterative scheme similar to EM or K-means that has 2 steps. One step will update the image intensity model based on the current segmentation. The second step will then update the segmentation based on the new image intensity model. Does this improve the results?

7 What to submit

Submitting Your Data to the Competition

Pick the method/parameters that yielded best cross-validation accuracy in the above experiments and apply it to the segmentation of the test dataset. Save the results as NIFTI files, named subxxx_seg.nii. These files should have the same structure as the segmentation files in the training directory, i.e., each voxel should have a single label. Make a zip file with these images and submit as part of your assignment.

Submit:

1. The **.zip** file with your segmentations.

2. All your MATLAB source code. Make sure that it is documented, so a reader can follow what you did.
3. A short report explaining what you did and a table of the results for each segmentation algorithm.

8 How the assignment will be graded

1. Your ability to complete correctly all the parts
2. The complexity and nature of any improvements over the core algorithms as provided for ICM or Graph Cut and as described in the papers for the Random Walker segmentation.
3. Any extra work done
4. The quality of the report and submitted code.

The top ranked team for this part of the challenge will receive an extra 7 points towards the overall assignment grade.