Graph Cuts for Markov Random Fields (MRF)

Henrik Aanæs

May 22, 2008

The aim of this exercise is to give you an introductory practical experience with using graph cuts for solving MRF problems in image analysis. The exercise is to be run from MatLab (with the use of some supplied C++ functions, linked to via mex).

1 Demo and setup

In order to solve the graph cut problems, a wrapper for the C++ functions found at:
`http://www.adastral.ucl.ac.uk/~ vladkolm/software.html`
have been supplied.[1] Binary mex files for windows and unix are also provided. In order to compile and get used to the package a short M-file is supplied, i.e. `GraphCutDemo.m`.
Tasks:

1. Run `GraphCutDemo.m`. If there is problems compiling please contact a/your TA.
2. Make a drawing of the graph it is running on.
3. Convince yourself that the obtained result is the correct one.

2 Binary Segmentation

Here you should segment an image of a human brain 'slice', which is available in the file `brain.mat`. You should segment it into two classes, fluid and brain matter (simply use the distribution for CSF and Grey matter, later on in the exercise brain matter is split into two categories).

Convince yourself that you understand how the prior (the weights for the terminal edges) is derived from a simple clique potential.

[1] Please note the copy right notice

	CSF(fluid)	Grey Matter	White Matter
Mean ($\mu(K)$)	32	47	55
Standard deviation ($\sqrt{\Sigma}_K$).	5	4	3

Table 1: Pixel distributions for the image classes.

The label of site (i, j) is denoted f_{ij} and the associated observed pixel value is denoted y_{ij}. The posterior probability is given by

$$P\left(f_{ij} = K | f_{kl} \in \mathcal{N}_{ij}, y_{ij}\right) = c(T) \exp\left(\frac{-U(y_{ij}|f_{ij} = K) + \beta n_{ij}(K)}{T}\right) \quad . \qquad (1)$$

Where $U(y_{ij}|f_{ij} = K)$ is derived from the Mahalnanobis distance, and is given by:

$$U(y_{ij}|f_{ij} = K) = \frac{\log(\det(\Sigma_K)) + (y_{ij} - \mu(K))^T \, \Sigma_K^{-1} \, (y_{ij} - \mu(K))}{2} \quad . \qquad (2)$$

You don't need to compute $c(T)$, since it is the same for all class labels K. Notice that the classes have different determinants. The values for the distributions are given in Table 1. For now you should only use the classes 'CSF fluid' and 'Gray Matter'.

Tasks:

1. Write up (2) for the two classes.
2. Compute a segmentation, via graph cuts, based only on one-cliques, i.e. $n_{ij}(K)$ is set to zero or omitted. Thus you should only supply terminal edges to the graph. Comment on the result.
3. Incorporate neighborhood two clique terms i.e. use the $n_{ij}(K)$. You can get an idea on how to do that from the supplementary slides provided on the webpage. Comment on the result, when varying the strength of the neighborhood cost, $n(k)$, (a good starting value is 1).

3 Multi Label Segmentation, via α-Expansion

Here we should segment the brain image into three classes, using graph cuts and alpha expansion.

Tasks:

1. Write it as a separate MatLab function called `Init`, which should be used for initializing the α-expansion iterative algorithm. This should work similar to task two above, in that each pixel should be classified as the most probable class based *only* on its pixel value, i.e. no neighborhood cost, $n(K)$. Note that this should not necessarily be done via graph cuts.

2. Write a MatLab function that performs one α-expansion move for a given label. Test the function for all three labels, with the output from `Init` as initialization.

3. Segment the brain image into three classes using alpha expansions, i.e. a succession of alpha expansion moves. How many moves do you need for 'reasonable' convergence? Comment on the result, when varying the strength of the neighborhood cost, $n(k)$, (a good starting value is 1).

Hand in a small 'report' with concise answers to the above raised issues, along with figure illustrating your results, and any additional observations you may have made.