Problem Set 3: Markov Random Fields

The purpose of this problem set is to implement an MRF for image segmentation. This will resemble the MRF discussed in class, in which there is a node for each pixel. Each node can be labelled either foreground or background. There is also an observation at each pixel. The intensity at each pixel is generated from a distribution that depends on the label. Finally, pixel nodes are connected in a 4-connected grid, so that each pixel has four neighbors, and the cliques of the MRF have a size of at most 2.

You can test your code using the test image on the class web page. I have also included images showing the labels produced by each part of the problem set.

For each problem, include images showing the results and your code.

1. Data term (potentials for cliques of size 1).

   First, we just create a clique potential to represent the relationship between each node and the observation at that node. Assume that background pixel intensities are drawn from a Gaussian distribution with a mean of 128 and $\sigma = 30$. Assume that each foreground pixel is drawn from a mixture of two Gaussian distributions. Half the time, it is drawn from a Gaussian with a mean of 30, and half the time from a mean of 225. Both distributions have $\sigma = 30$.

   Write code to compute the clique potentials of size one that are appropriate for these distributions. Using these potentials, find the MAP estimate of all the labels.

2. Pairwise terms

   Next, add a clique potential for cliques of size two that captures the fact that neighboring pixels tend to have the same label. If two pixels have the same label, the value of the clique potential is zero. If they have different labels, the clique potential is $K = .897$. Find the MAP estimate of the labels for the test image, using the Iterated Conditional Modes (ICM) algorithm. With this algorithm, one loops through all the labels, adjusting each label to the value that has greatest probability, given that all other labels are fixed. This is continued until a complete pass through all the labels does not change any of their values.

3. Simulated Annealing

   Next, implement optimization by simulated annealing. In this approach we proceed by:

```
for T = Temperatures
   for i = 1:NumberRepetitions
      index = RandomIndexOfALabel
      f = current labels
      f' = current labels with label(index) changed
      p = P(f')/P(f);
      % P(f) indicates probability of labeling f
      if rand < p
        change label(index) to new value
        % rand is Matlab rand; # from uniform distribution between 0 and 1
      end
   end
end
```

To replicate the results on the web page, use a cooling schedule in which Temperatures = [64 32 16 8 4 2 1 .5 .25 .125 .0625 .03125 eps] and NumberRepetitions = 81920. Feel free to try other parameters to obtain better results.

4. Conditional Random Fields

   Now try using a CRF. We will make the pairwise clique potentials depend on the gradient at each location. The clique potential for neighboring pixels with the same label will still be 0. However now, when the gradient in the direction between the pixels is large, the potential will be small if the pixels have different labels, but the potential will be large if the gradient is small. We compute the gradient after smoothing with a Gaussian with a $\sigma = 1.5$.

   So, for example, if I is the image, consider the pixels I(i,j) and I(i+1,j). When these have the same label, the clique potential is 0. When they have different lablels, the clique potential is: $f(\frac{\partial I}{\partial x}(i,j)))$, where $f(x)$ is a function that goes to 0 when $abs(x)$ is large, and becomes big when $x = 0$. To obtain the results shown on the class web page, set M to be equal to the magnitude of the largest derivative in the x or y direction in the image. Then, use a clique potential of: $4 * K * (M - \frac{\partial I}{\partial x}(i,j))/M$ (and a similar potential for derivatives in the y direction).

5. Extensions

   Even with a pretty contrived test image, the results so far aren't that great, and I wasn't able to get good results with this approach on even a simple real image. Try out at least one extension to the current MRF, and try to apply it to a real image to get good segmentation results. You might also want to alter parameters or try out slightly different energy functions. Here are some suggested options:

   (a) With only two labels, one can find the global optimum for the CRF above using graph cuts. Implement this. Does it improve performance on the test images, or on other images you can try? (You do not need to implement your own min cut algorithm, you can use existing code).

   (b) For the data term in Problem 1, try allowing the user to specify portions of the foreground and background, and then learn a distribution suitable for the current image. (Note that these first two suggestions lead to a segmentation algorithm similar to Boykov and Jolly).

   (c) Extend the CRF method in part 4. Make the clique potential dependent on color or texture changes.

   (d) Or, explore your own ideas.

   Include the results of your method on at least one real image of your choice. Describe briefly what you implemented. Then, give one or two paragraphs explaining whether your extensions seemed to produce good results, and why.