Computer exercise 6
Image segmentation using Gaussian mixture models
Spatial Statistics and Image Analysis, TMS016

1 Introduction

The purpose of this computer exercise is to give an introduction to image segmentation using Gaussian mixture models. When in doubt about how to use a specific function in Matlab, use `help` and `doc` to get more information.

2 Segmentation of satellite data

Throughout the exercise, we will use a satellite image of Gothenburg as an example, which you can download from the course homepage.

- Use the K-means algorithm to classify the image with different number of classes K and plot the results. Can you get one class that only contains the water with this method? An implementation of the K-means algorithm is given in the function `normmix_kmeans`.
- Choose some value of K (such as 3) and use `normmix_sgd` to estimate a general Gaussian mixture model to the data. Then use `normmix_classify` to classify the image and plot the results.
- `normmix_classify` classifies the pixels by choosing the class with the highest probability for each pixel. The function also returns the actual class probabilities as a second output. To investigate the uncertainty of the classification, use the function `classification2rgb`, which you wrote in the first exercise, to plot the probabilities. Depending on how you wrote the function, you might have to update it to allow for plotting of probabilities.
- Compare the results to the K-means classification, and in particular look at the estimated parameters of the mixture model and compare with those estimated by the K-means algorithm.

3 Plotting a segmented image

- Choose your favourite classification from above and plot the parts of the image as segmented by the method. As an example, for $K = 4$, you can plot the different parts using

```
I = imread('gothenburg_satellite.png');
I = double(I)/255;
Istack = reshape(I,[size(I,1)*size(I,2) size(I,3)]);
figure(1)
for k=1:4
  I_class = Istack;
  I_class(cl~=k,:)=256;
  subplot(2,2,k)
  imagesc(reshape(I_class,[size(I,1) size(I,2) 3]));axis image;
end
```

  where `cl` is the classification obtained from `normmix_classify` or `normmix_kmeans`.

4 Segmentation using relative colors

An often undesired feature of classifying the image using the RGB colors is that shadows can affect the results.

- Transform the image to relative colors (as in Exercise 1), or to LAB colors using rgb2lab, and classify the image using K-means. Compare with the classification using RGB colors.
- Perform the classification using the general Gaussian mixture model can compare with the classification using RGB colors.
- Which method, which color space, and which value of K, would you say works best for this image?

5 Morphological operations and improved segmentation

A problem with segmenting an image using a mixture model is that the intensity of the background may have a clear spatial trend. We can therefore improve the segmentation by first estimating the background, removing it, and then segmenting the updated image.

Load the rice.png image which is supplied with Matlab by writing **I = imread('rice.png');** and convert it to double values.

- Create a binary image by segmenting the image into two classes based on the greyscale values using a GMM
- Compute the image erosion, dilation, and opening using a circular structure element with different radii r and plot the results. A circular structure element can be constructed using **strel('disk',r),** and you can compute the erosion, dilation, and opening using **imerode**, **imdilate**, and **imopen**, respectively. Make sure that you understand what these operations are doing.
- Use **imopen** with a circular structure element to estimate the background of the grayscale image. Plot the image opening for different values of r to find a suitable value.
- Subtract the estimated background from the image and plot the results.
- Segment the image into two classes using a GMM and compare the result to the segmentation without the background removal.