SD 372 Pattern Recognition

Lab 3: Image Classification

1 Overview

The goal of this lab is to provide some context for classification in real-world situations. One particularly interesting application of classification is image classification, where content within an image is categorized into one of several classes. In this lab, image classification is performed on a set of texture images using both labelled classification (MICD) and unlabelled clustering (K-means).

Texture analysis is an important aspect of computer vision that depends on image classification. Examples include interpretation of remotely sensed images, determining depth cues from scenes, and identification of pertinent objects in medical imaging. In order to assist theoretical development and to be able to compare results from different studies, a standard set of texture images, the Brodatz images, is commonly used in research literature.

The necessary Matlab scripts, features, and image files for this lab can be found in a Zip file on the course homepage:
http://ocho.uwaterloo.ca/∼pfieguth/Teaching/372/sd372.html

For this lab, ten texture image files (*.im) are used. The texture images can be read and viewed in Matlab using the following commands:

```
image = readim('cloth.im') ;
imagesc(image) ;
colormap(gray) ;
```

2 Feature Analysis

Each texture image is either 256 × 256 or 256 × 128 pixels in size. We will select sixteen $n \times n$ blocks from each image. Let $d_{ij}(\alpha, \beta)$ be the gray level value of pixel (α, β) in the j^{th} block of the i^{th} image. We propose two simple features which measure the variability of each image in the horizontal and vertical directions:

$$\underline{x}_{ij} = \begin{bmatrix} \sum_{\alpha=1}^{n} \sum_{\beta=1}^{n-1} \left(d_{ij}(\alpha,\beta) - d_{ij}(\alpha,\beta+1)\right)^2 / ((n)(n-1)) \\ \sum_{\alpha=1}^{n-1} \sum_{\beta=1}^{n} \left(d_{ij}(\alpha,\beta) - d_{ij}(\alpha+1,\beta)\right)^2 / ((n)(n-1)) \end{bmatrix} \tag{1}$$

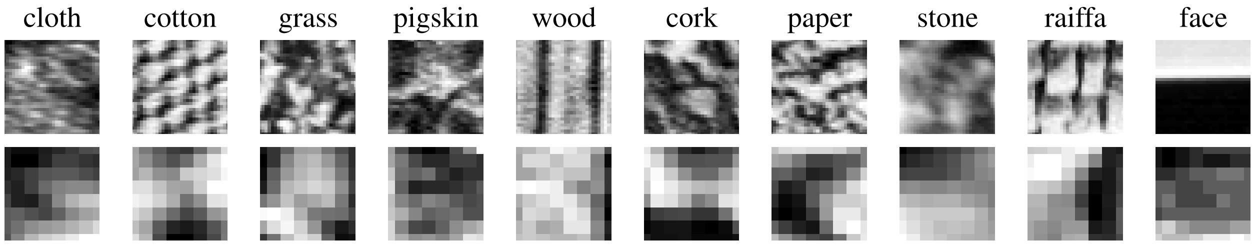

Figure 1: Sample image blocks from which features are derived. The top row consists of 32×32 blocks and the bottom row 8×8 blocks for each texture respectively. Clearly the textures are much harder to discriminate by eye based on the smaller 8×8 feature blocks, so we would expect a classifier trained from these features to have a poorer performance.

Luckily, all of the features have been pre-calculated and provided in the 'feat.mat' Matlab data file. The data file contains three feature matrices **f2**, **f8**, and **f32** which contain the features calculated from each of the 16 blocks for all 10 images for n=2, 8, and 32 respectively. Separate matrices **f2t**, **f8t**, **f32t** are provided as test data. Each column in each matrix represents one image block and is arranged out as follows:

$$\begin{bmatrix} \underline{x}_{ij} \\ i \\ j \end{bmatrix} \tag{2}$$

where $\underline{x}_{ij}$ is the feature vector in image i at block j.

To plot the features for a particular feature set, use the following commands:

```
load feat.mat
aplot(f2);
```

You will notice that the feature points are plotted using letters. Each letter corresponds to a different texture in the following order: cloth, cotton, grass, pigskin, wood, cork, paper, stone, raiffa, face. Sample blocks from each texture image are shown in Figure 1.

Keep in mind how our two features are defined in (1):

1. By looking at the texture images themselves, (e.g., on the SD372 home page), which do you think would be most likely to be confused with the other images? Why?
2. Which images are likely to be most distinct? Why?

Be sure to answer the questions in the context of the features extracted from the texture images.

	Classified						
Truth	1	2	3	...	8	9	10
1	14	1	0	...	1	0	0
2	1	15	0	...	0	0	0
3	0	0	13	...	0	1	0
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$	$\vdots$
8	1	0	0	...	11	1	0
9	0	0	1	...	1	12	0
10	0	0	0	...	0	0	16

Table 1: An example confusion matrix: In this example, Image 1 was classified correctly fourteen times, classified incorrectly as Image 2 once, etc.

3 Labelled Classification

1. For each of the three feature matrices **f2**, **f8** and **f32**, develop the MICD classifier. You are not required to plot the classification boundaries. Apply the classifier to the test data **f2t**, **f8t** and **f32t**.
2. What was the misclassification rate for each image and for each n? Prepare three confusion matrices, each like the one shown in Table 1, one table for each value of n, to compare how the images are classified.
3. Compare and explain the results for the different values of n.

4 Image Classification and Segmentation

In the 'feat.mat' data file are two large matrices, **multim** and **multf8**. The matrix **multim** contains a image that is composed of multiple different textures. The matrix **multf8** contains the corresponding feature vector for each pixel using $n = 8$. The matrix elements **multf8(i,j,1), multf8(i,j,2)** are the two features corresponding to row i, column j of the image.

Use your MICD classifier corresponding to $n = 8$ to classify each pixel in **multf8**. Create an array **cimage** such that **cimage(i,j)** contains the classified class number. Plot the result using

imagesc(cimage)

State your observations. How do the classified regions in the resulting classified image **cimage** compare to original texture image **multim**?

5 Unlabelled Clustering

For this section, we will perform unlabelled clustering to perform unsupervised image classification on the texture images we've classified earlier using the MICD classifier. For this section we will treat the data points, $\underline{x}_{ij}$, as *unlabelled* points in feature space. Implement the K-means algorithm using the features in the **f32** feature matrix only.

1. Assume $K = 10$. Pick ten points randomly from your data $\underline{x}_{ij}$ and use these ten points as your initial prototypes.
2. Apply the K-means algorithm using the selected prototypes. Plot the position of the 10 converged prototypes superimposed on the data $\underline{x}_{ij}$.
3. Now repeat step 2 except using the fuzzy K-means approach with $b = 2$.
4. Repeat steps 2 and 3 a few more times (with different, random initial prototypes each time). It is not necessary to submit any plots for these trials; however, compare the variability in the clustering from your trials. State your conclusions and observations. Furthermore, qualitatively compare your results from the unlabelled clustering with the results you obtained using the MICD classifier. How do they compare?

Take a look again at the data points in matrix multf8. These are unlabelled data – without looking at multim, we don't actually know which pixel location belongs to which class. How well do you think K-means would work on the data in multf8? Could we use unlabelled clustering to classify the image multim?

6 Report

Include in your report:

- A brief introduction.
- Discussion of your implementations and results.
- Printouts of pertinent graphs (properly labelled).
- M-files for each section.
- Include responses to all questions.
- A brief summary of your results with conclusions.