Pattern Recognition and Machine Learning

Task - 1: Character Recognition using Bayesian Classifier

Build a pattern recognizer by using Bayesian Classifier concepts.

Given

- 200 training images of 3 characters
- Size of each image is 128*128 pixels

- 100 test images of 3 characters
- Size of each image is 128*128 pixels

Assumptions

- Samples are generated from a multi-dimensional Gaussian distribution, having class specific mean vectors $\boldsymbol{\mu}_i$.

Concepts

- Supervised Learning
- Bayesian Decision Theory
- Maximum Likelihood Estimation
- Discriminant Functions, Decision Boundaries
- Joint Probability, Conditional Probability
- Linear Algebra
  - Matrix Multiplication
  - Finding inverse of a matrix

Discriminant Function:

$$g_i(\mathbf{x}) = \ln p(\mathbf{x}|\boldsymbol{\omega_i}) + \ln P(\boldsymbol{\omega_i}) \tag{1}$$

The above expression comes from the Bayes' Formula.

Multivariate Gaussian Distribution:

$$p(\mathbf{x}) = \frac{1}{(2\pi)^{\frac{d}{2}}|\boldsymbol{\Sigma}|^{\frac{1}{2}}} \exp\left[-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^t\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right] \tag{2}$$

where $\mathbf{x}$ = d-component column vector
$\boldsymbol{\mu}$ = d-component mean vector
$\boldsymbol{\Sigma}$ = d-by-d covariance matrix

For multivariate normal density, discriminant function is equal to

$$g_i(\mathbf{x}) = -\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu_i})^t|\boldsymbol{\Sigma}|^{-1}(\mathbf{x}-\boldsymbol{\mu}) - \frac{d}{2}\ln 2\pi - \frac{1}{2}\ln|\boldsymbol{\Sigma_i}| + \ln P(\boldsymbol{\omega}_i) \tag{3}$$

The additive terms which are equal for all the classes can be removed as it doesn't change the decision of classifier. Thus the 2nd term can be removed from the above expression.

There are 3 different ways using which we can model the covariance matrix:

Case 1 - Arbitrary covariance matrix for each class
Case 2 - Common covariance matrix for all classes
Case 3 - Identity covariance matrix for all classes (also called Naive Bayes classifier)

The above 3 cases are implemented using Python and the images are classified with the discriminant function. The image is assigned to the class which has the highest value of discriminant function.

Mean and covariance matrices are estimated using the Maximum Likelihood techniques. The size of images are reduced to 32*32 pixels to avoid memory issues while storing the covariance matrix. A regularization term of the form $\lambda\mathbf{I}$ was added to beat the curse of dimensionality.

Summary of accuracies (in percentages):

	Class 1	**Class 2**	**Class 3**
Case 1	95	89	97
Case 2	93	91	97
Case 3	86	86	100

Python functions and modules used:

- Tkinter and tkFileDialog for opening file dialog.
    - askdirectory()
- scipy for processing images
    - scipy.misc.imread()
    - scipy.misc.imresize()
- numpy for dealing with arrays containing pixel intensities
    - hstack()
    - matmul()
    - linalg.inv(), linalg.det()
- some other list functions