

Assignment 3. Feature Selection, Linear Classifiers, Dimensionality Reduction**Due Date: 11:59pm Sun Dec 8, 2013****Grading: 4404 assignments are marked out of 110. 5327 assignments are marked out of 120.**

Please submit your assignment report electronically as a pdf file called a3report.pdf via Moodle. Your report should be brief and well organized, with figures properly formatted and captioned. Please make sure that you answer each question and show each plot requested.

Part 1. Short Answer Question (10 marks)

This part of the assignment will help to develop and demonstrate your understanding of the theory underlying mixture models and EM. Please use clear notation and define all terms when deriving results.

The update equation for the covariances of an MVN mixture model is:

$$\Sigma_k = \frac{1}{N_k} \sum_{n=1}^N r_{nk} (\mathbf{x}_n - \mu_k)(\mathbf{x}_n - \mu_k)^T, \text{ where the responsibilities } r_{nk} \text{ are given by}$$

$$r_{nk} = \frac{N(\mathbf{x}_n | \mu_k, \Sigma_k) \pi_k(t)}{\sum_{k=1}^K N(\mathbf{x}_n | \mu_k, \Sigma_k) \pi_k(t)}, \text{ and } N_k \triangleq \sum_{n=1}^N r_{nk}.$$

Please derive this equation. (Hint: recall that

$$\frac{d}{dA} |A| = |A| A^{-T}, \quad \frac{d}{dA} \mathbf{x}^T A \mathbf{x} = \mathbf{x} \mathbf{x}^T \text{ and } \frac{d}{dA} A^{-1} = -A^{-2} \text{ for square, invertible matrix } A.)$$

Part 2. Programming Questions (4404: 100 marks, 5327: 110 marks)

In this assignment we will explore **dimensionality reduction methods**, and implement, train and evaluate a number of different classifiers. The problem we address is **frontal face detection**.

The training dataset is available from the Assignment 3 section of our Moodle home. The training dataset consists of a **3,480 x 361 input data matrix and a 3,480 x 1 input target vector**. Each row of the data matrix represents a **19 x 19 pixel image**. (To visualize the nth row of the data matrix x as an image use the MATLAB command `imn=reshape(x(n,:),19,19);`). **Each entry of the target vector is either +1 to indicate a face or -1 to indicate a non-face**. There are **1,215 faces and 2,265 non-faces in the training dataset**. You can assume the **same proportion of faces and non-faces in the test dataset**.

Note that each of the images has been pre-normalized to have 0 mean and unit variance. This turns out to be important, as the original images vary a lot in luminance and contrast, but these variations are not predictive of the presence of a face.

In addition to your report you will also submit a MATLAB source code file (*.m file). Please make sure this file satisfies the stated naming and input/output requirements exactly as specified below (Question 12), as evaluation will be automatic. You may also submit a .mat parameter file used by your MATLAB function, and you may assume that it will be in the current directory when your function is evaluated. All three files should be submitted as a single compressed file.

Feature Selection (10 marks)

1. **(10 marks)** Calculate the **signal-to-noise ratio for each pixel** of the training images. Plot the result as an image. Use the **colorbar function to interpret the image in terms of SNR**. What regions of the face appear to be most important for detection? What regions are least important?

Principal Components (35 marks)

2. **(7 marks)** Plot a scatterplot of the intensities of the first 2 pixels for the face dataset and also for the non-face dataset. How would you describe the statistical relationship between these two pixel intensities? What does this predict about the first principal component?
3. **(7 marks)** Compute the first **5 principal components** of the face and non-face datasets, and display them as images. How would you describe them?
4. **(7 marks)** Plot the eigenvalues of the covariance matrix in decreasing order of magnitude for both face and non-face datasets.
5. **(7 marks)** Compute the principal components for the face and non-face dataset separately. Plot the **proportion of variance explained as a function of the number of principal components used**. How many components must you use in order to account for 95% of the variance in each of the datasets?
6. **(7 marks)** Select a random face from the dataset and use the principal components of the face data to show:
 - a. An approximation based upon the first principal component
 - b. An approximation based upon the first two principal components
 - c. An approximation based upon the first five principal components
 - d. An approximation based upon the first ten principal components

Classifiers (4404: 45 marks, 5327: 55 marks)

Please summarize your cross-validation results for the following questions in a well-organized table. Indicate error rate, not proportion correct.

7. **(10 marks)** Build a Bayesian classifier, assuming the **class-conditional distributions are multivariate normal with different means and different but diagonal covariances**. Using a leave-one-out cross-validation method, estimate the error rate for this classifier.
8. **(10 marks)** Build a least-squares linear classifier. Using a leave-one-out cross-validation method, estimate the error rate for this classifier. Code the algorithm directly (do not use third-party code, MATLAB classification functions, etc.).
9. **(CSE 5327 Only: 10 marks)** Build a logistic regression classifier using the iterative reweighted least squares algorithm. Using 100-fold cross-validation, estimate the error rate for this classifier. Code the algorithm directly (do not use third-party code, MATLAB classification functions, etc.).

10. **(10 marks)** Using the `svmlib` package, build an SVM classifier. Use the default options for training, except for the kernel: use `'-t 0'` to select a linear kernel function. Using 100-fold cross-validation, estimate the error rate for this classifier.
11. **(15 marks)** Repeat Q7, but this time using two separate positive and negative M -dimensional PCA subspaces based upon the positive and negative training data, respectively, where $M = 75$. Evaluate the error using the leave-one-out cross-validation method. How does this method compare to your other methods? (Note that this method is not perfectly valid probabilistically, as we are no longer using a common feature space for the two classes. However, in practice this method can work well.)

Competition (10 marks)

12. For this question, you will try to build the classifier you think will perform best on the reserved test dataset. You can assume that the proportion of faces and non-faces in the test set will be the same as in the training set. You may use one of the classifiers you constructed in Questions 7-11, or you may try additional strategies. Once you have selected the classifier, submit it through Moodle as a MATLAB function. The function will be called `a3q12.m`, and will accept one parameter: the $N \times D$ data matrix for the test dataset. It will return one parameter: your predictions for the $N \times 1$ target label vector for the test dataset. The coding should be as for the training dataset: +1 for face, -1 for non-face.

Please provide a clear description of the method you are submitting, including the values of any free parameters, and list results for your method (called `a3q12`) in the cross-validation table you produced above.

You may also submit an external `.mat` parameter file, which you can assume resides in the current directory when your function is evaluated. Please submit both files, along with your report, as a single compressed file.

The student with the lowest error rate on the test dataset will win a prize!

Some things you could try:

1. Try varying the covariance assumptions for your Bayesian MVN classifier. You could try estimating the full covariance matrices, or try assuming isotropic covariance matrices. You can also vary these parameters in your subspace methods.
2. Optimize the dimensionality of the subspaces used in Q11.
3. Different kernels and parameters for your SVM classifier.
4. Different normalization techniques. For the classifiers constructed in Q7-10, what happens if you whiten the data first?
5. Different methods for dimensionality reduction, e.g.,
 - a. Subsample the image
 - b. Select the subset of pixels with highest SNR
6. Use a Gaussian mixture model for the conditional distributions, estimated using EM.
7. A combination of the above techniques