Hong Kong University of Science and Technology
COMP 5212: Machine Learning
Fall 2013

Project 1
Due: 17/10/2013, Thursday, 11:59pm

1 Objectives

The objectives of this project are three-fold:

1. To acquire a better understanding of dimensionality reduction by implementing two classical methods, one unsupervised and one supervised.

2. To explore two simple extensions of the dimensionality reduction methods for addressing the singularity problem.

3. To compare the performance of different dimensionality reduction methods for classification by conducting empirical comparative study on two real-world data sets.

2 Major Tasks

The project consists of the following tasks:

1. To implement principal component analysis (PCA) for unsupervised dimensionality reduction.

2. To implement linear discriminant analysis (LDA) for supervised dimensionality reduction.

3. To implement two extensions, called regularized LDA (RLDA) and PCA+LDA, for supervised dimensionality reduction.

4. To implement a Bayesian classifier based on the parametric approach.

5. To conduct empirical study to compare different dimensionality reduction methods for classification.

6. To write up a project report.

Each of these tasks will be elaborated in the following subsections.

2.1 PCA and LDA

Your PCA implementation is expected to allow dimensionality reduction from the input dimensionality d to any value of $k \in \{1, \ldots, d\}$ by accepting k as a user-specified parameter. For the LDA implementation, the maximum value of k allowed is $K-1$ where K is the number of classes in the data.

You are expected to implement PCA and LDA all by yourself so you can gain a better understanding of these two classical dimensionality reduction methods. Although there exist implementations that you can call as functions to perform PCA and LDA, you should not use them for this project. However, you may use built-in functions provided by Octave/MATLAB, e.g., for eigendecomposition.

Octave/MATLAB is the preferred language choice which can allow you to do fast prototyping possibly at the expense of run-time efficiency. You may also use some other programming languages such as C++ and Java if you insist, but this is not recommended because you then cannot take advantage of the powerful and convenient matrix manipulation capabilities and built-in functions provided by Octave/MATLAB.

2.2 Two Extensions: RLDA and PCA+LDA

LDA involves solving the following generalized eigenvalue problem:

$$\mathbf{S}_B\mathbf{w} = \lambda\mathbf{S}_W\mathbf{w}.$$

To solve this problem, the within-class scatter matrix $\mathbf{S}_W$ must be nonsingular. For applications involving high-dimensional data in which the sample size is relatively small, it is not uncommon for $\mathbf{S}_W$ to be singular.

RLDA provides a simple solution to overcome the singularity problem by adding a regularization parameter $\alpha > 0$ to each diagonal element of $\mathbf{S}_W$, i.e., by replacing $\mathbf{S}_W$ with $\mathbf{S}_W + \alpha\mathbf{I}$. Since the ordinary LDA can be regarded as a special case of RLDA with $\alpha = 0$, it suffices to have a single implementation for both LDA and RLDA.

Another simple approach, which may be referred to as PCA+LDA, is to first apply PCA to reduce the dimensionality of the data before LDA is applied on the lower-dimensional representation of the data. Obviously the above implementation for PCA and LDA can be used directly to realize this hybrid method.

2.3 Parametric Classifier

You are to implement a parametric classifier for multi-class classification. You may take the assumption that each class follows a normal distribution. However, your implementation is expected to allow for univariate or multivariate normal distribution of any dimensionality $k \in \{1, \ldots, d\}$.

Based on the maximum likelihood estimation (MLE) approach to parameter estimation, you can estimate the prior class probabilities, class means and class covariance matrices (or variances for $k = 1$) from data to implement the classifier using appropriately defined discriminant functions.

Caution should be taken in computing the inverse of a covariance matrix when the covariance matrix is singular. This is particularly the case when no dimensionality reduction is performed so that the feature dimensionality is high. You are recommended to use the `pinv` function (for computing pseudoinverse) instead of the `inv` function.

2.4 Empirical Study

You will use two data sets for this empirical study. The data can be downloaded from here:

`http://www.cad.zju.edu.cn/home/dengcai/Data/MLData.html`

The first data set is a subset of the MNIST database of handwritten digits. Specifically, the 10k training set

`http://www.cad.zju.edu.cn/home/dengcai/Data/MNIST/10kTrain.mat`

and 10k test set

`http://www.cad.zju.edu.cn/home/dengcai/Data/MNIST/Test.mat`

will be used. Note that the files are in binary format to keep their sizes smaller. You may use the `load` command in Octave/MATLAB to load each file. Once a file is loaded, you may use the `whos` command to see the matrix variables `fea` and `gnd` thus created for the features and labels, respectively. Each example is described by $28 \times 28 = 784$ features and a class label which indicates its membership in one of the 10 digit classes. Note that the feature values are in the range 0 to 255. You may normalize them to the range $[0, 1]$. The labels take values in the range 0 to 9. Depending on your implementation, you may change them to the range 1 to 10. Other than these, no further preprocessing of the data is needed for this project. Note that the test data (both features and labels) should not be used for classifier training. They are used for measuring the prediction accuracy of the trained classifier on the unseen test data.

Your comparative study will include the different settings below before performing classification using the parametric classifier:

1. No dimensionality reduction performed on the data
2. PCA performed on the data with $k \in \{1, 2 \ldots, 9\}$
3. RLDA performed on the data with $k \in \{1, 2 \ldots, 9\}$ and $\alpha \in \{10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 1\}$
4. PCA+LDA performed on the data with $k \in \{20, 40, 60, 80, 100\}$ for the PCA stage and $k \in \{1, 2 \ldots, 9\}$ for the LDA stage

The second data set you will use is the COIL20 database of object images. All examples in the data set are stored in a single file:

`http://www.cad.zju.edu.cn/home/dengcai/Data/COIL20/COIL20.mat`

To simulate the situation with limited training data, you will use only 12 training examples per object class. Specifically, the examples with row indices $6n + 1,\ 0 \leq n < 240$ will be used for classifier training while the rest will be used as test data.

The different settings for your comparative study are as follows:

1. No dimensionality reduction performed on the data
2. PCA performed on the data with $k \in \{1, 2 \ldots, 11\}$

3. RLDA performed on the data with $k \in \{1, 2 \ldots, 11\}$ and $\alpha \in \{10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 1\}$

4. PCA+LDA performed on the data with $k \in \{40, 60, 80, 100\}$ for the PCA stage and $k \in \{1, 2 \ldots, 11\}$ for the LDA stage

Note that for PCA/LDA/RLDA, the transformation is computed based on the training data only. The transformation computed will then be applied to the test data (without having to perform PCA/LDA/RLDA again) during the testing phase.

The training data, with or without performing dimensionality reduction, is used to estimate the parameters of the classifier. You are expected to measure the classification accuracy separately for the training set and the test set. The program should be written in such a way that the TA can run it easily to verify the classification results on the training set and the test set reported by you.

2.5 Report Writing

In your report, you only need to present the experimental results. Besides reporting the classification accuracy (for both training and test data) in numbers, graphical aids should also be used to compare the performance of different methods visually.

3 Some Programming Tips

As is always the case, good programming practices should be applied when coding your program. Below are some common ones but they are by no means complete:

- Using functions to structure program clearly
- Using meaningful variable and function names to improve readability
- Using indentation
- Using consistent styles
- Including concise but informative comments

For Octave/MATLAB in particular, you are highly recommended to take full advantage of the built-in functions (e.g., `mean`, `cov` and `max`). Also, using loops to index individual elements in matrices and arrays should be avoided as much as possible. Instead, block indexing without explicitly using loops is much more efficient. Proper use of these implementation tricks often leads to speedup by orders of magnitude.

4 Project Submission

Project submission should be done electronically using the Course Assignment Submission System (CASS):

`http://cssystem.cse.ust.hk/UGuides/cass/student.html`

There should be two files in your submission:

1. **Project report** (with filename `report`): preferably in PDF format.
2. **Source code and a README file** (with filename `code`): all necessary code for running your program as well as a brief user guide for the TA to run the program easily to verify your results, all compressed into a single ZIP or RAR file. The data should not be submitted to keep the file size small.

When multiple versions with the same filename are submitted, only the latest version according to the timestamp will be used for grading.

5 Grading Scheme

This project will be counted towards 10% of your final course grade.

6 Academic Integrity

Please read carefully the HKUST Academic Honor Code on the course website.

While you may discuss with your fellow classmates on general ideas about the project, your submission should be based on your own independent effort. In case you seek help from any person or reference (from the Web or other sources), you should state it very clearly in your submission. Failure to do so is considered plagiarism which will be handled with appropriate disciplinary actions.