Experiment 2
Multi-Category Classification Using Binary Linear Classifiers

1. Introduction and Objective

In this experiment, we investigate a technique for multi-category classification based on binary classifications. The technique is then applied to Fisher's 3-class datasets of Iris plants to demonstrate its effectiveness. The dataset of Iris plants to be used in this experiment was created and published in 1936 by R. A. Fisher [R1]. Fisher's paper is a classic in the field and is referenced frequently to this day, as a matter of fact the dataset is arguably the best-known in the pattern recognition literature [R2]. The dataset includes features of 150 Iris plants of 3 species known as Setosa, Versicolor, and Virginica, where each sample Iris is represented by a 4-dimensional vector in terms of lengths and widths of the sepal and petal of the flower.

In this experiment, the above data set was divided into two sets, one for training and the other for testing. The training data set includes 120 samples where there are 40 samples for each species. The test data set includes 30 samples and there are 10 samples for each species.

References

[R1] R. A. Fisher, "The use of multiple measurements in taxonomic problems", *Annual Eugenics*, vol. 7, part II, pp. 179-188, 1936.

[R2] UCI Machine Learning, http://archive.ics.uci.edu/ml, University of California Irvine, School of Information and Computer Science.

2. Background

2.1 The idea of multi-category classification using linear binary classifiers

Consider a labeled dataset $\mathcal{D}$ that contains a total of K data classes with $K > 2$, namely,

$$\mathcal{D} = \mathcal{D}_1 \bigcup \mathcal{D}_2 \bigcup \cdots \bigcup \mathcal{D}_K$$

The technique is known as *one-versus-the-rest*, and it is built on a simple process of converting the data classes at hand into a *two-class* scenario so that binary classification becomes applicable. The process is run K times, each time the original datasets are grouped into a "*positive class*" and a "*negative class*" appropriately so as to identify an optimal linear function for classifying these two classes. The K linear functions so obtained are then applied to the test data for multi-category classification.

To start, we single out class $\mathcal{D}_1$ and assign label $y_n = 1$ to all its samples and re-name the data class as "positive class" $\mathcal{P}$. And we combine the rest of the classes into *one* class and assign label $y_n = -1$ to all its samples and call it "negative class" $\mathcal{N}$. That is,

$$\begin{cases} \mathcal{P} = \mathcal{D}_1 \text{ with } y_n \equiv 1 \\ \mathcal{N} = \mathcal{D}_2 \bigcup \cdots \bigcup \mathcal{D}_K \text{ with } y_n \equiv -1 \end{cases}$$

In this way, one deals with a two-class dataset so that a linear model $f_1(\boldsymbol{x}, \boldsymbol{w}_1, b_1) = \boldsymbol{w}_1^T \boldsymbol{x} + b_1$ is obtained for binary classification by doing its best to separate class $\mathcal{P}$ from class $\mathcal{N}$.

Next, the process described above is repeated in that we single out class $\mathcal{D}_2$ and assign label y_n = 1 to all its samples and re-name it as "positive class" $\mathcal{P}$, while combining the rest of the classes into *one* class and assign label $y_n = -1$ to all its samples and call it "negative class" $\mathcal{N}$. A linear model $f_2(\boldsymbol{x}, \boldsymbol{w}_2, b_2) = \boldsymbol{w}_2^T \boldsymbol{x} + b_2$ is obtained based on the updated data sets $\mathcal{P}$ and $\mathcal{N}$.

The process continues in a similar fashion until K such linear models $f_i(\boldsymbol{x}, \boldsymbol{w}_i, b_i) = \boldsymbol{w}_i^T \boldsymbol{x} + b_i$ for i = 1, 2, …, K are obtained. Classification of a new sample $\boldsymbol{x}$ outside the dataset is then performed by the classifier below:

$$\boldsymbol{x} \text{ belongs to class } i^* \text{ if } f_{i^*}(\boldsymbol{x}, \boldsymbol{w}_{i^*}, b_{i^*}) \text{ reaches maximum among } \{ f_i(\boldsymbol{x}, \boldsymbol{w}_i, b_i) \text{ for } i = 1, 2, ..., K \} \quad \text{(E2.1)}$$

2.2 Linear classifier for binary classification

The linear model for binary (i.e. two-class) classification was studied in Sec. 1.5, see pages 24-26 of the course notes. Let the two data classes $\mathcal{P}$ and $\mathcal{N}$ assume the form

$$\mathcal{P} = \{(\boldsymbol{x}_i^{(P)}, 1),\ i = 1, 2, ..., N_p\} \text{ and } \mathcal{N} = \{(\boldsymbol{x}_i^{(N)}, -1),\ i = 1, 2, ..., N_n\}$$

respectively.

From Eq. (1.34) of the course notes, it follows that the optimal parameters of linear function $f(\boldsymbol{x}, \boldsymbol{w}, b) = \boldsymbol{w}^T \boldsymbol{x} + b$ that best separate class P from class N can be found by solving the system of linear equations

$$\hat{\boldsymbol{X}}^T \hat{\boldsymbol{X}} \hat{\boldsymbol{w}} = \hat{\boldsymbol{X}}^T \boldsymbol{y} \quad \text{(E2.2a)}$$

where

$$\hat{\boldsymbol{w}} = \begin{bmatrix} \boldsymbol{w} \\ b \end{bmatrix}, \quad \hat{\boldsymbol{X}} = \begin{bmatrix} \boldsymbol{x}_1^{(P)T} & 1 \\ \vdots & \vdots \\ \boldsymbol{x}_{N_p}^{(P)T} & 1 \\ \boldsymbol{x}_1^{(N)T} & 1 \\ \vdots & \vdots \\ \boldsymbol{x}_{N_n}^{(N)T} & 1 \end{bmatrix}, \quad \boldsymbol{y} = \begin{bmatrix} 1 \\ \vdots \\ 1 \\ -1 \\ \vdots \\ -1 \end{bmatrix} \quad \text{(E2.2b)}$$

・ Note that in this experiment parameter $\boldsymbol{w}$ is a vector of length 4, and b is a scalar; the sizes of classes $\mathcal{P}$ and $\mathcal{N}$ are set to N_p = 40 and N_n = 80; and $\boldsymbol{y}$ is a constant vector of length 120, with its first 40 components being 1 and the rest of 80 components being –1.

2.3 The dataset

We have divided Fisher's dataset into two sets which are available from the course website as **`D_iris_tr.mat and D_iris_te.mat`**. To be precise, **`D_iris_tr`** is a matrix of size 4×120 where its first 40 columns are associated with Iris Setosa, the next 40 columns are for Iris Versicolor, and the last 40 columns are for Iris Virginica. **`D_iris_te`** is a matrix of size 4×30 where its first 10 columns are associated with Iris Setosa, the next 10 columns are for Iris Versicolor, and the last 10 columns are for Iris Virginica.

Once **`D_iris_tr.mat and D_iris_te.mat`** are downloaded, the code below prepares the datasets.

```
Xtr1 = D_iris_tr(:,1:40);
Xtr2 = D_iris_tr(:,41:80);
Xtr3 = D_iris_tr(:,81:120);
```

```
Xte1 = D_iris_te(:,1:10);
Xte2 = D_iris_te(:,11:20);
Xte3 = D_iris_te(:,21:30);
```

• Data sets **Xtr1**, **Xtr2**, and **Xtr3** are train data from Setosa, Versicolor, and Virginica, respectively, each contains 40 samples. Data sets **Xte1**, **Xte2**, and **Xte3** are test data from Setosa, Versicolor, and Virginica, respectively, each contains 10 samples.

3. Procedure

3.1 From the course website download data matrix **D_iris_tr.mat and D_iris_te.mat**

3.2 Follow Sec. 2.3 above to prepare training and testing datasets.

3.3 Based on the material in Secs. 2.1 and 2.2 above to prepare MATLAB code to carry out the procedure below. Note that in this lab experiment the number of classes is $K = 3$. Consequently, 3 binary classifications are required to produce linear models

$$\begin{cases} f_1(\boldsymbol{x}, \boldsymbol{w}_1, b_1) = \boldsymbol{w}_1^T \boldsymbol{x} + b_1 \\ f_2(\boldsymbol{x}, \boldsymbol{w}_2, b_2) = \boldsymbol{w}_2^T \boldsymbol{x} + b_2 \\ f_3(\boldsymbol{x}, \boldsymbol{w}_3, b_3) = \boldsymbol{w}_3^T \boldsymbol{x} + b_3 \end{cases}$$

Also note that vector $\boldsymbol{y}$ in (E2.2b) is a constant vector of size 120 $\times$ 1 throughout the procedure, and is given by

```
y = [ones(40,1); -ones(80,1)];
```

(i) Use dataset **Xtr1** as class $\mathcal{P}$ and datasets **Xtr2**, and **Xtr3** combined as class $\mathcal{N}$ to prepare matrix $\hat{\boldsymbol{X}}$. Compute optimal parameters $\boldsymbol{w}_1$ and b_1 by solving the equation in (E2.2a). This allows to construct a linear model $f_1(\boldsymbol{x}, \boldsymbol{w}_1, b_1)$.

(ii) To obtain the second linear model, repeat Step (i) above, where classes $\mathcal{P}$ and $\mathcal{N}$ need to be re-constructed, see Sec. 2.1 above. This will let you prepare correct matrix $\hat{\boldsymbol{X}}$ and hence a correct model $f_2(\boldsymbol{x}, \boldsymbol{w}_2, b_2)$.

(iii) Repeat Step (ii) above with appropriately re-constructed classes $\mathcal{P}$ and $\mathcal{N}$ to establish the third linear model $f_3(\boldsymbol{x}, \boldsymbol{w}_3, b_3)$.

(iv) Apply the classifier in (E2.1) to the test data sets {**Xte1**, **Xte2**, **Xte3**} by performing the following steps:

Step 0: Set counter `mis_class = 0;` Assign label $y_k = 1$ to all samples in **Xte1**; label $y_k = 2$ to all samples in **Xte2**; and label $y_k = 3$ to all samples of **Xte3**.

Step 1: For a given test sample $\boldsymbol{x}$, apply the 3 linear models obtained above. This yields three real values $f_i = \boldsymbol{w}_i^T \boldsymbol{x} + b_i$ for $i = 1, 2, 3$. Then identify index i^* such that f_{i^*} is the largest among $\{f_1, f_2, f_3\}$. Classify sample $\boldsymbol{x}$ to class i^*. To record your result, it is recommended to use a 3-component column vector whose i^*th component is set to value 1 while the other two components are set to value 0. For the kth test sample, call this vector $\boldsymbol{e}_k$. In

addition, compare value i^* with the label of sample $\boldsymbol{x}$ being tested, if they are not equal, add 1 to counter **`mis_class`**.

Step 2: Repeat Step 1 to cover the entire test dataset. This should yield a total of 30 vectors $\{\boldsymbol{e}_k,\ k = 1, 2, \ldots, 30\}$ and **`mis_class`** tells you the total number of misclassifications made. If you combine all 30 $\{\boldsymbol{e}_k\}$ to form a matrix

$$\boldsymbol{E} = \begin{bmatrix} \boldsymbol{e}_1 & \boldsymbol{e}_2 & \cdots & \boldsymbol{e}_{30} \end{bmatrix}_{3\times 30}$$

then $\boldsymbol{E}$ provides sufficient information to allow you to produce a 3×3 confusion matrix (see Example 1.11 of the course notes) that summarizes the performance of the classification technique applied.

Report the confusion matrix obtained as well as the error rate which is given by **`mis_class/30`** in this case.

Include your MATLAB code in the lab report.