MACHINE LEARNING FROM DATA
Fall 2018

Lab Session 3 – K-Nearest Neighbors and Parzen windows

1. Goal

The goal of this session is to

- Use and compare two non-parametric classifiers, k-nearest neighbors and Parzen windows.
- Test these methods on two real datasets, an image dataset of handwritten digits and a small high dimensional microarray dataset.
- Implement a cross-validation strategy for hyperparameter selection.

2. Instructions

Getting the material:

- Download and uncompress the file ML_Lab3_soft.zip

Handling your work:

- Answer the questions in the document Lab3_report_yourname.pdf
- Save the report and Matlab code (if necessary) in a folder, and upload the compressed folder (zip, rar).

3. Introduction and previous study

Read the slides corresponding to lecture 4: non parametric classifiers.

In this session we will use two datasets, Zip-code and Microarray. These are examples of classification problems where the number of samples is relatively small compared to the dimensionality of the feature space and the number of classes.

We will use two parametric models: k-nearest neighbors, where we will have to choose an optimal value for the parameter k, and Parzen windows, where we will have to select an optimal value for the parameter h.

4. Zip-code dataset

4.1. Characteristics of the dataset

The Zip code dataset can be found in this repository:
https://web.stanford.edu/~hastie/ElemStatLearn/data.html

Each element in the dataset is a vector of d=256 features corresponding to the intensity values of a 16x16 image of a handwritten digit. Images are vectorized row by row. There are 10 classes corresponding to the 10 digits (0, …, 9). The dataset is already split into training and test subsets. The training set contains 7291 samples and the test set contains 2007 samples.

This is the information provided by the authors:

```
Normalized handwritten digits, automatically scanned from envelopes by the
U.S. Postal Service. The original scanned digits are binary and of different
sizes and orientations; the images here have been deslanted and size
normalized, resulting in 16 x 16 gray scale images (Le Cun et al., 1990).

The data are in two zipped files, and each line consists of the digit id (0-
9) followed by the 256 grayscale values.
```

```
There are 7291 training observations and 2007 test observations, distributed
as follows:
         0    1   2   3   4   5   6   7   8   9 Total
Train 1194 1005 731 658 652 556 664 645 542 644 7291
 Test  359  264 198 166 200 160 170 147 166 177 2007

or as proportions:
         0    1   2    3    4    5    6    7    8    9
Train 0.16 0.14 0.1 0.09 0.09 0.08 0.09 0.09 0.07 0.09
 Test 0.18 0.13 0.1 0.08 0.10 0.08 0.08 0.07 0.08 0.09

Alternatively, the training data are available as separate files per digit
(and hence without the digit identifier in each row)

The test set is notoriously "difficult", and a 2.5% error rate is excellent.
These data were kindly made available by the neural network group at AT&T
research labs (thanks to Yann Le Cunn).
```

First, we will work with data in the original format (d=256). Then we will apply PCA and MDA to reduce the dimensionality of the feature space, working with vectors of size d'=64 (for PCA) and d'=9 (for PCA and MDA).

Next, we will apply k-fold cross validation to select the most appropriate hyperparameter value (that is k for kNN and h for Parzen windows).

4.2. Classification using kNN

Run the script `add_path_db.m` to update Matlab path.
Read the script `lab3_zip.m` identifying the different sections in the code.

Run the script `lab3_zip.m` with the option 'no reduction' to keep vectors with all 256 features. You can maximize figures for a better visualization of the images.

Answer the following questions:

Q1. Complete a table with training and test errors obtained for kNN and discuss the results. What is the value of *k*? Analyze the confusion matrices and identify the two most challenging classes.

Q2. Run again the script, using PCA to reduce the dimensionality of the feature space, selecting d'=64 features. Observe the eigenvectors and the images reconstructed using only the first d' eigenvectors (those with the highest eigenvalues). Discuss. Complete a table with training and test errors. Discuss the results and compare with the previous case (no PCA).

Q3. Repeat the previous analysis using PCA with d'=9 features, and MDA with d'=9 features. Discuss which method is the best for image reconstruction and which one is preferable for classification.

4.3. kNN with cross validation

In this section we will use MDA with d'=9 features.

However, instead of using an arbitrary value of *k* in kNN, we will use a method called K-fold cross validation in order to select the optimal value of this parameter.

Q4. Edit the script and modify the code to find the optimal value of *k* by K-fold cross validation on the training set. Use K=10 folds. Plot the average train and validation errors (average over the folds) as a function of *k*. Use the optimal value of *k* to compute the error on the test set.
Hint: use the function `cvpartition.m` to partition the dataset.

Q5. Copy the new code

4.4. Parzen windows with cross validation

Now we will use Parzen windows to classify the zip-code dataset.
Use the function `predict_parzen.m` for making predictions:

```
Predict_test = predict_parzen(X_train,Labels_train,N_classes,h,X_test)
```

Again, reduce dimensionality to d'=9 features with MDA.

Q6. Use K-fold cross validation (K=10) to select the best parameter *h*, where possible values of *h* are 1, 10, 20 and 100.
Plot the average train and validation errors (average over the folds) as a function of K. Use the optimal value of *h* to compute the error on the test set.

You will notice that this method is computationally very slow. If it takes too long you can try with a lower value of K (for example K=2).

Q7 Copy the new code

5. Microarray dataset

5.1. Characteristics of the dataset

The Microarray dataset can be found in this repository:
https://web.stanford.edu/~hastie/ElemStatLearn/data.html

Each element is a vector of d=6830 features representing genetic information contained in cells from c=14 different organs affected by cancer.
Each feature is a real value between -10 and +10 (approximately). Each feature represents the presence (+10) or absence (-10) of one of 6830 genes. A dataset sample corresponds to an organ affected by cancer and different samples correspond to different subjects. There are 64 samples in the dataset. The number of classes is 14 (cancer types)
The name of the dataset refers to the technique used to extract this genetic information.

The main problem with this dataset is the large dimensionality of vectors, which impedes the use of many of the classification models studied in this course. Note, for instance, that estimated covariance matrices would be singular or rank deficient. Therefore, this dataset will only be classified using k-nearest neighbor.

This is the information provided by the authors:

```
NCI microarray data
```

```
Source and reference:
http://www-stat.stanford.edu/~tibs/ElemStatLearn/data.html

NCI microarray data

6830 genes
missing values have been imputed via SVD
60 cell lines, labels are below

1: CNS         5 samples
2: RENAL       7 samples
3: BREAST    9 samples
4: NSCLC        9 samples
5: UNKNOWN     1 samples
6: OVARIAN     6 samples
7: MELANOMA  8 samples
8: PROSTATE    2 samples
9: LEUKEMIA  6 samples
10:K562B-repro 1 samples
11:K562A-repro 1 samples
12:COLON       7 samples
13:MCF7A-repro 1 samples
14:MCF7D-repro 1 samples
```

5.1. Classification using kNN

Run the script lab3_MA.m
Read the script and identify the main code sections.
The script loads a dataset and selects only the classes with at least two samples per class.
You have to enter the value of the parameter `k_neig`.

Q8. Complete a table with the training and test errors for different values of k=1, 2, 3, 4. Discuss the results.