COSC522 Machine Learning Project 3 - Dimensionality Reduction and Performance Evaluation

Yangsong Gu

October 8, 2021

1 Introduction

The objective of this project is, first of all, to practice the usage of dimensionality reduction as one of the preprocessing steps and how to draw an ROC curve and confusion matrix as part of the post-processing (performance evaluation) steps. You need to use both supervised dimensionality reduction method (i.e., FLD) and unsupervised method (i.e., PCA) for that purpose. We will continue using pima as the dataset but in a more comprehensive way. The second objective is to extend the horizon of machine learning applications and solve a seemingly quite unrelated problem – image compression. Think hard if image compression should be solved using supervised learning or unsupervised learning, and what are the features in this application.

2 Data sets

Two datasets will be used. The first is the Pima Indians dataset with 7 features. The second is a beautiful color image of flowers.

- The Pima dataset: pima.tr (the training set) and pima.te (the test set). Preprocessing you need to do: (Please note the change of terminology used!)
  - 1) Remove the first row with any text editor; Change the labels from "Yes" and "No" to "0" and "1" respectively indicating 'with disease' and 'without disease' - you can do this using the same text editor; (Note that intuitively "Yes" should be 1 and "No" be 0. So both labeling schemes would be fine.
  - Standardize Normalize the data set to make the features comparable (or with the same scale). Suppose x is a data sample, μ_i is the mean of each feature i, $sigma_i$ is the standard deviation of each feature i, then standardization normalization is conducted by $(x - \mu_i)/\sigma_i$. Keep in mind that you also need to standardize normalize the samples in the test set. Be careful which mean and standard deviation you should use. (For each sample in the test set, use the same $_i$ and σ_i you derived from the training set.)
- The flower image
  The image given is a 120x120 full-color image. It can also be treated as a 120 x 120 x 3 matrix, representing a 3-dimensional feature space of 120 x 120 samples.

3 Performance Metric

Besides the three metrics used in Projects 12, i.e., 1) overall classification accuracy, 2) classwise classification accuracy, and 3) run time, we'll introduce a fourth metric that measures the quality of the compressed image as compared to the original full-color image - 4) root mean squared error (RMSE). We'll also add confusion matrices and ROC curves to performance evaluation.

4 Task 1

4.1 Task 1.1

Implement both supervised and unsupervised dimensionality approaches (FLD and PCA). Only the pima dataset is used.

Some notations:

- **nX**: the standardized dataset.
- **fX**: the projected data using FLD.
- **pX1**: the projected data using PCA.
- **pX**: the projected data using PCA(with error rate samller than 15

How much the error rate would be when keeping one component?

The error rate is computed by:

$$\epsilon = 1 - \frac{\sum_i^k \lambda_i}{\sum_j^d \lambda_j}$$

where ϵ denotes the error rate, superscript k is the number of component removed, d is dimension of data set. λ is the eigenvalue. Note that the eigenvalue is sorted in a descending order when computing the error rate.

Firstly, PCA was implemented on training set nX. When keeping one component, the error rate is **65.58%**. The corresponding projection vector computed from training set nX is:

$$v_1 = [-0.3655, -0.3660, -0.4126, -0.4315, -0.3912, -0.0291, -0.4713]^T$$

The corresponding eigenvalue is 2.4093.

How many dimensions need to be kept to ensure a at-most 15% error rate?
Likewise, project matrix was obtained from the training set and applied to project the test set. To meet at-most 15% error rate, we need to at least keep **5** components. The corresponding eigenvalue from largest to smallest are:

$$\lambda = [2.4093, 1.4964, 0.9119, 0.8000, 0.6901]$$

Comparing the histogram of fX and pX1 on training set
Figure 1 presents the distribution of projection of training set.1 where the left side shows the projection result from FLD and the right side shows the result from PCA. The two histograms suggest that the data sample do not follow the Gaussian distribution, as the left one left-shewed and right one is concave in the middle. Thus, the parametric learning might be misleading. Instead, non-parametric might be a better option.

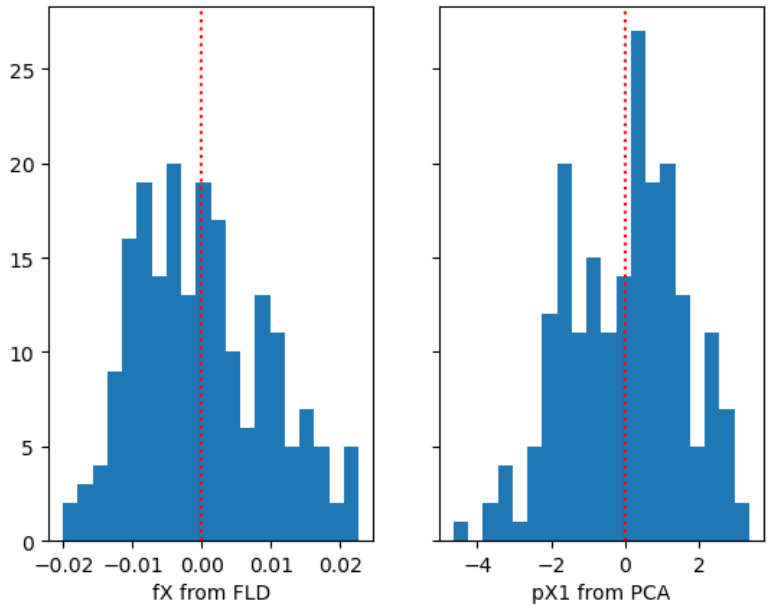

Figure 1: projection of training set, left: FLD, right; PCA

4.1.1 Task 1.2

Apply both Case 3 and kNN on nX, fX, px1, and pX. Use the "k" where you obtain the best performance on nX in Project 2.

Performance metrics

Table 1 and 2 conclude the performance metrics of case 3 and Knn on different test set, assuming the 1:3 prior probability (class 0 vs. class 1).

- Focusing on Case 3.
    1. fX from FLD increased overall and class 1 accuracy compared against raw training set. In orther words, FLD dimentionality reduction not only reduced the raw Dimension, but also improved the model accuracy. In addition, results from fX outperforms other projection (raw) data set in terms of overall and class 1 accuracy.
    2. the above comparison indicates that the supervised dimensionality reduction (FLD) outperforms the unsupervised dimensitonality approach (PCA).
    3. Comparing PCA with one component (pX1) and 5 components (pX), pX that has lower error rate did come up with a higher overall accuracy and class 1 accuracy than pX1.
    4. Similar to fX, pX not only reduced the dimensionality but also increase the performance, with overall accuracy and class wise accuracy are all higher than nX.
- Focusing on Knn (k=12)

1. The projection data decrease the accuracy of overall and class-wise accuracy. pX1 generates the worst overall and class-wise accuracy. while the results of FLD (fX) and PCA (pX) were slightly affected by the dimensionality reduction.

- Comparing Case 3 and Knn

1. the unsupervised learning model (Knn) shows the higher accuracy on nX than supervised learning model(Case 3).
2. when they are performed on projection data, the advantages also differ.

Table 1: Model performance of Case 3

	Overall acc.	Acc. of class 0	Acc. of class 1	Run time (seconds)
nX	76.81%	54.13%	87.89%	0.0588
fX	80.42%	54.13%	93.27%	0.0090
pX1	73.19%	34.86%	91.93%	0.0055
pX	77.41%	57.80%	87.00%	0.0409

Table 2: Model performance of kNN (k=12)

	Overall acc.	Acc. of class 0	Acc. of class 1	Run time (seconds)
nX	78.01%	57.70%	87.98%	1.0248
fX	78.01%	64.22%	84.75%	0.6163
pX1	70.18%	49.54%	80.27%	0.6050
pX	76.51%	56.88%	86.10%	0.8725

Table 3: Confusion matrix of Case 3

	P	N
P	59	50
N	27	196

(a) Case 3 on nX

	P	N
P	59	50
N	15	208

(b) Case 3 on fX

	P	N
P	38	71
N	18	205

(c) Case 3 on pX1

	P	N
P	63	46
N	29	194

(d) Case 3 on pX

Table 4: Confusion matrix of Knn (k=12)

	P	N
P	63	46
N	27	196

(a) Knn on nX

	P	N
P	70	39
N	34	189

(b) Knn on fX

	P	N
P	54	55
N	44	179

(c) Knn on pX1

	P	N
P	62	47
N	31	192

(d) Knn on pX

Confusion matrices

Table 3 and 4 present the confusion matrix generated by case 3 and Knn, respectively.

ROC curve of pX

Figure 2 displays the ROC curve where the horizontal axis shows the False positive rate and vertical axis shows the True positive rate.

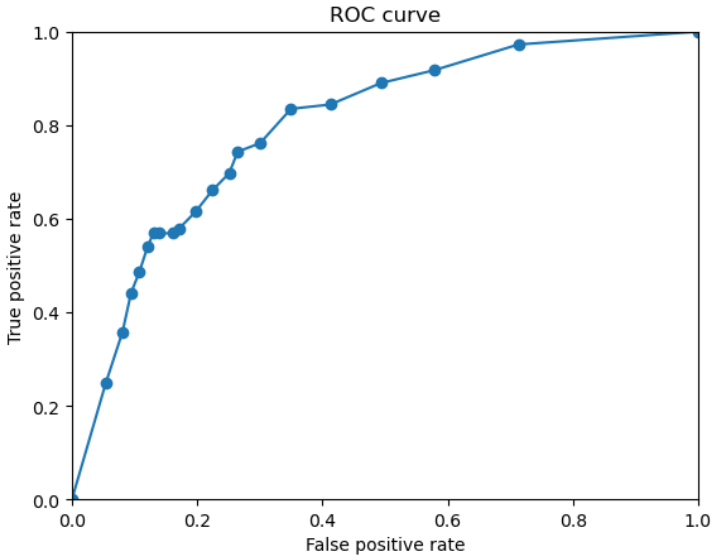

Figure 2: ROC curve of Case 3

5 Task 2

Use kmeans and wta you implemented in Project 2 to solve the image compression problem. Each pixel of this color image has three components: red, green, and blue. Each component is an 8-bit unsigned char. That is, each pixel is represented using 24 bits, a total of 2^{24} possible colors. You are asked to use less number of bits to represent each pixel. For example, if you want to only use 256 colors to represent the original full-color image, then you are basically only using 8 bits to represent each pixel, instead of 24 bits. We refer to the color image not showing its full-color potential as pseudo-color image. Figure3 shows the raw flower picture.

Figure 3: Raw picture

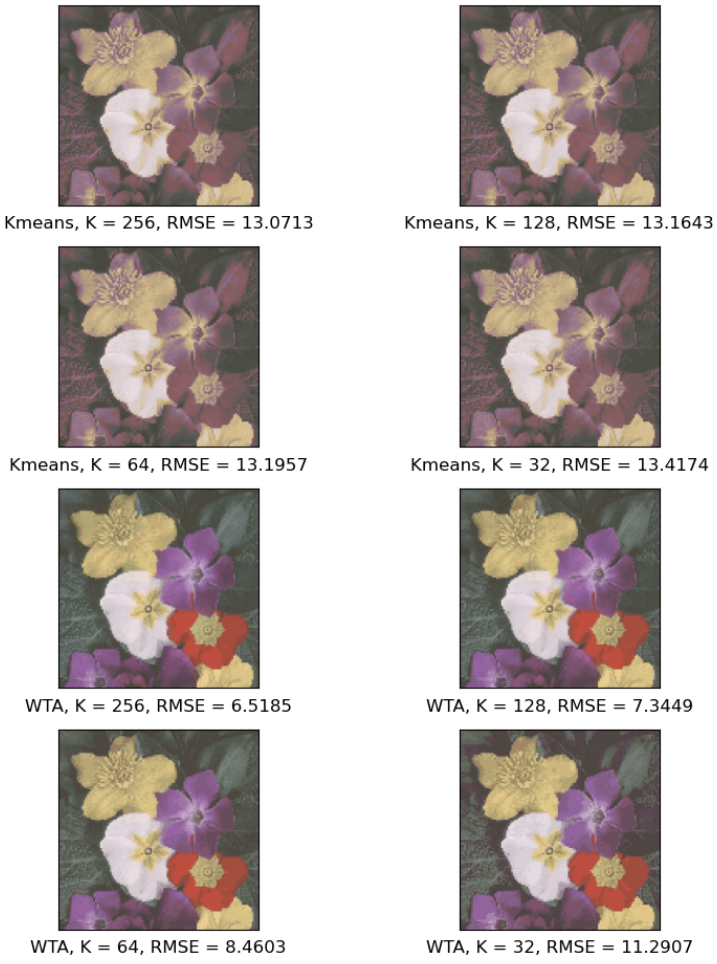

Figure 4: Image process, Kmeans and Wta

5.1 Task 2.1

Draw a table with 8 rows and 2 columns showing the generated pseudocolor images with k = 256, 128, 64, and 32 different colors using kmeans and wta. Underneath each image, display the reconstruction error measured in terms of RMSE.

Figure 4 demonstrates the new image compressed by Kmeans and wta, respectively. Top 4 subplots show the results computed by kmeans and bottom 4 subplots show the results constructed by wta.

5.2 Task 2.2

Comment on the results both visually and through quantitative measurement (i.e., RMSE).

- Compared with raw figure 3, Kmeans detect fewer types of color than wta, with red petal and green leaf misclassified as dark purple color. In this regard, all of wta results significantly

retain color feature of raw picture.

- In Kmeans approach, the error (RMSE) slightly varies as K changes from 32 to 256. In contrast, the wta results seem more sensitive to clsuter number K. We can see that RMSE of K = 32 is 5 more than K = 256.
- For each K, wta result produced much less error.
- wta costs much more computation time than kmeans (See table 5).
- The compressed image from Kmeans retains fewer color than wta, which suggests that kmeans approach is prone to stuck in local optimum.

Table 5: Run time (seconds)

K	Kmeans	wta
256	28.89	413.69
128	15.84	172.49
64	8.79	81.85
32	5.40	30.95

6 Final discussion

Dimensionality reduction is a critical pre-processing step in dealing with high dimensional problems. In this project, two typical dimensionality reduction approaches were implemented to reduce dimensionality for further clustering purpose. FLD is known as the supervised learning while PCA is known as unsupervised learning. **Firstly**, FLD and PCA were used to preprocess the pima data set which has 7 dimensions. Case 3 and Knn were employed on both raw and projection data, the results show that 1). for the supervised classification algorithm (i.e., Case 3), the compressed data of don't significantly decrease the model performance, instead, the fX from FLD shows a higher accuracy than full data set in case 3. 2) for the unsupervised learning approach (Knn), the accuracy decreased slightly after dimensionality reduction. Those two major findings suggest that PCA and FLD effective ways to remain the raw data features while not hurting the model accuracy a lot. **Secondly**, two unsupervised clsutering algorithm were employed to the image reconstruction problem. I found that kmeans generates the higher error than wta under the same clusters. The results indicate that Kmeans approach is prone to stuck in the local optimum while the on-line learning approach (wta) could potentially skip this problem by a subtle learning rate. Comparatively speaking, although wta cost more time than kmeans, it retains relatively good features of raw picture.

7 Code

Please see the attachment.