The University of Adelaide, School of Computer Science

Introduction to Statistical Machine Learning

Semester 2, 2020 Assignment 3: Implementation of PCA and k-means algorithms

DUE: 4 Nov. 2020, Wednesday 11:55 PM

Submission

Instructions and submission guidelines:

• You must sign an assessment declaration coversheet to submit with your assignment.

• Submit your assignment via the Canvas MyUni.

PCA and k-means

You are required to write code for performing PCA on a given dataset. After performing PCA, you are required to (1) use various classifiers to perform classification and (2) implement k-means clustering to group data.

Data

You will use a subsampled version of the MNIST dataset to test your model. It contains images for 10 digits (10 classes). The training set contains 6,000 samples and the test set contains 1,000 samples.

The images from the data set have the size 28 x 28. They are saved in the csv data files mnist_train.csv and mnist_test.csv.

Every line of these files consists of an image, i.e. 785 numbers between 0 and 1.

The first number of each line is the label, i.e. the digit which is depicted in the image. The following 784 numbers are the pixels of the 28 x 28 image.

Tasks

Specifically, the tasks you need to complete are:

- 1. Write code for learning PCA parameters, i.e., mean vector and project matrix from the training data
- 2. Apply PCA to both training and testing set. Then perform classification with the 1-nearest neighbour classifier. Analyse the performance change against different reduced dimensions. (suggestion: from 256 to 10)

- 3. Write code for implementing k-means clustering. Apply k-means clustering to the MNIST training set without dimensionality reduction. Plot the loss curve, that is, the change of loss value of k-means algorithm with respect to the number of iterations.
- 4. Randomly choose one training sample from each class as initial clustering centres (so in total 10 centres). Performing k-means to group data into 10 groups with those initialized centres. For each cluster, calculate the percentage of samples sharing the same digit as the initial group centre. Average those percentages as an evaluation metric for k-means clustering. Repeat the above experiment with dimensionality reduced features and calculate the average percentage again. Note that you keep the initial clustering centres fixed through out those experiments. Analyse the performance change against different reduced dimensions. (suggestion: from 256 to 10)
- 5. [Master student only] Append 256 noisy dimensions to the original data, that is, for each sample xi, appending a 256-dimensional random feature vector to xi to make it a 1040-dimensional feature. Then repeat the PCA classification experiments by using 1-nearest classifier and a linear SVM as classifiers. Test and analyse the results. The following lines give an example code (Matlab) for appending noise features
    - % Let X be the original data
    - [N,d] = size(X);
    - R = randn(N,d); % using Gaussian noise in this experiment
    - X_ = [X,R];

Requirements:

You can choose either Matlab, Python, or C/C++. I would personally suggest Matlab or Python.

The PCA and k-means part of your code should not rely on any 3rd-party toolbox. Only Matlab's built-in API's or Python/ C/C++'s standard libraries (numpy, pandas) are allowed. However, you can use 3rd-party implementation of linear SVM for your experiments.

You are also required to submit a report (<10 pages in PDF format), which should have the following sections (report contributes 50% to the mark; code 50%):

• An algorithmic description of PCA. (5%)

• An algorithmic description of k-means (5%)

• For task 2, some analyses of your implementation. You should plot an error curve against the number of reduced dimensions. (10% for master students and 20% for undergraduate students)

• For task 3 and 4, some analyses of your result. For task 3, you should plot the loss curve. For task 4, you should plot an average percentage curve against the number of reduced dimensions. (10% for master students and 20% for undergraduate students)

• For task 5, you should present and analyse the results of adding noisy dimensions before performing PCA. (for example, the impact of choosing different reduced dimensions and classifiers, whether or not PCA is sensitive to noisy dimensions and why) (20% for master students only)

In summary, you need to submit (1) the code for the above tasks and (2) a report in PDF.