ARI 510 Lab 2: Unsupervised Learning and Feature Selection

University of Michigan-Flint

Fall 2024

(See course Canvas page for due date)

1 Overview

The goal of this assignment is to introduce you to unsupervised learning techniques, including dimensionality reduction and clustering, and to explore how these techniques can inform feature selection in machine learning. You will work with the **Human Activity Recognition Using Smartphones Dataset**, which contains measurements from smartphone sensors (accelerometers and gyroscopes) recorded while participants performed various activities (e.g., walking, sitting, standing).

The dataset consists of 561 features extracted from raw sensor data. Your task will be to explore the dataset, reduce its dimensionality using **Principal Component Analysis (PCA)**, perform clustering, and use the insights gained from unsupervised learning to perform feature selection. By the end of this assignment, you should be able to identify the most important features and compare the performance of a machine learning model trained with selected features versus the full feature set.

You can download the dataset from the UCI Machine Learning Repository at the following link: Human Activity Recognition Using Smartphones Dataset

2 Instructions

Follow the steps below to complete the assignment.

Step 1: Standardization

After downloading the dataset, load the training set X (features) and y (target label). Standardize the features so that each has a mean of 0 and a standard deviation of 1. You might want to use something like the StandardScaler class from sklearn. Some of the steps later (i.e., PCA and clustering) work better with standardized data.

Step 2: Exploratory Data Analysis (EDA)

Perform an initial exploratory analysis of the dataset. This is always a good idea to do when working with a new dataset so you get a good feel for what you are working with before you start training any models.

- Visualize the distribution of the target variable (the activity labels). A histogram is a good idea here and there are a few ways to achieve this. If you are using pandas, for example, you can use this method or you can use a visualization library like seaborn (or any other one you like).
- Apply **PCA** to the dataset (just the features, do not include the labels) to reduce its dimensionality.
- Plot the **explained variance ratio** for each principal component. You can use the attribute `explained_variance_ratio_` from the sklearn PCA object to easily get this. Use a line plot where the x axis is the component (starting with the first principal component) and the y axis represents the explained variance ratio. Try to identify the "elbow" of this plot where the values begin to level out and no longer decrease as sharply. The number of components that correspond to this "elbow" is your value for k, the number of components you want to use in a reduced-dimensionality version of your data. Keep this value in mind for later.
- Visualize the dataset in the 2D space using only the first two principal components (using a scatter plot). Color the data points based on their activity labels to observe any natural groupings.

Step 3: Clustering

- Apply **K-Means clustering** to the dataset (only the features X, and using the full set of features). Use the the **silhouette score** to determine the optimal number of clusters. The idea here will be to try different numbers of clusters, e.g., try the integers from 2 to 10, when you run the k-means algorithm. Each time, record the silhouette scores and try to identify the number of clusters that gave you the highest average silhouette score. You don't need to create any visualizations here but you may choose to.
- Visualize the clusters using the top two principal components. That is, make a 2D scatter plot like you did before, but use the cluster labels to color the datapoints instead of the ground truth activity labels. Are the clusters well-separated? Does the clustering align with the activity labels?
- Repeat the clustering analysis using **DBSCAN** and compare the results to K-Means. Note that DBSCAN does not require the selection of the number of clusters as this is determined as part of the clustering algorithm itself.

Step 4: Feature Selection

Use these two methods to modify your original set of features. You will select only a set of k features in both situations so you will compare models using the same number of features.

- Select the top k principal components of the PCA-transformed data you created earlier.
- **Feature Importances from a Tree-Based Model**: Train a Random Forest model and examine the feature importance scores. Select the top k most important features (check out the attribute `feature_importances_` from the sklearn`RandomForestClassifier` class.

Step 5: Supervised Learning with Selected Features

- Train a simple supervised learning model (e.g., Logistic Regression or SVM, using sklearn) on the full feature set and evaluate its performance on the test set (this should be the first time you use the test set!) using accuracy, precision, recall, and F1-score. You can use the default hyperparameters for the model if you like.
- Now, train the same model using only the selected features from the previous step. Compare the performance of the model with the reduced feature sets to the performance with all features.
- Finally, write your report based on the outline provided in the next section.

3 Deliverables

You should submit:

- A written report (PDF) and
- Code used for your experiments with a short README file that explains how to run the code

If you want to combine these, you can submit a PDF version of a Jupyter notebook which mixes python code, visualizations and text. You may also include your report as part of a github README.md file in markdown with links to the code in the same repository.

3.1 Report

The report should include the following sections:

1. **Exploratory Analysis** including a histogram of the class distribution, plot of explained variance ratio for PCA, and a 2D scatter plot using the first two PCs and colored by class label. With each figure, write a brief

description of what the figure tells us about this specific dataset (i.e., do not just say "a histogram shows the distribution of the data", talk about what *this* histogram tells us about *this* dataset). Make sure to include what your value of k is for the PCA-transformed data that you plan to use later.

2. **Clustering** including details about your silhouette analysis and at least 2 visualizations: one for K-means and one for DBSCAN. Describe what you observed as you used these two methods.

3. **Classification** show the results of your (at least) 3 variations of the classification model. One using the full feature set and the other 2 using reduced feature sets from the Feature Selection step. Reflect on the results. Did the model's performance change? How did reducing the number of features impact accuracy, interpretability, and computational efficiency?

4. **Reflection** Write a paragraph to reflect on the overall process you just went through. You might answer questions like: what was the most challenging part? How did you overcome any challenges? What is your main takeaway from completing this assignment? Did anything surprise you?

3.2 Code

Make sure the code is organized and readable such that it is easy to know which parts of the code correspond to each part of the assignment. If you used any non-standard libraries (things other than core Python lbiraries, sklearn, or common data visualization libraries) please make a note of how these should be installed.

4 Other questions?

Please feel free to ask on Discord in the #homework-help channel.