CSCI 4350 - Open Lab 4

Unsupervised Learning

Overview

Develop a software agent in Python to perform K-means clustering on labeled classification data.

Procedure

1. Create a Python program (kmeans.py) which calculates a K-means clustering of a provided set of input training data, assigns classification labels to each cluster using a majority vote, and then reports the classification performance on a separate set of input testing data.
    - The program should take **4 command-line arguments** (integer: random seed, integer: the number of clusters, string: input training data filename, string: input testing data filename)
    - The program should read in the **training data** file (one training example per line, see below)
    - The program should read in the **testing data** file (one testing example per line, see below)
    - Each line will contain any number of real feature values and a single interger value (the class label) at the end
    - The program should perform a K-means clustering by first **initializing** K centroid vectors (no class labels included) using K random examples from the training data
    - The program should then determine the **closest** vector to each training example (using Euclidean distance), and create a new set of vectors by **averaging** the feature vectors of the closest training examples.
    - The program should **repeat** the previous step **until** the centroid vectors no longer change (i.e. until all training examples are assigned to the same vector on two consequtive iterations)
    - Once the mean cluster vectors have been calculated, a class label will be assigned to each vector by taking a majority vote amongst it's assigned examples from the training set (ties will be broken by preferring the smallest integer class label).
    - Finally, the program will calculate the closest vector to each testing example and determine if the cluster label and the testing example label match (a correct classification).
    - The program should then output the number of testing examples classified **correctly** by the K-means clustering
2. Use your program to calculate the **mean** performance and **standard error** of the K-means classifier using leave-10-out repeated random subsampling cross-validation.
    - Use your program to determine the performance of K-means across all numbers of clusters **nClust**=[1,2,3,...,N-10]
    - For each in nClust, run 100 random suffles of the data (link below) with a training set size of N-10 and a test set size of 10 (use **split.bash/parallelize.bash** to help as needed).
    - Make a plot of mean performance (as a percenage) vs. number of clusters which includes error bars of +/- 1.96 standard errors away from the mean.
3. Write a report (at least 2 pages, single spaced, 12 point font, 1 inch margins, no more than four pages) describing the K-means method, the code you developled to implement it, the performance of the code under cross-validation (using the statistics above for justification), any limitations of the overall approach, and describe any additional implementation details that improved the performance of your code.

Requirements

- You should utilize the Iris data set to build and test your K-means agent (download: iris-data.txt)
- A link to the original data set, with additional information can be found here: Iris@UCI
- **DO NOT** use the original data set from the UCI link as input; I have re-formatted it to my specifications
- You should also utilize the Breast Cancer data set to analyze the performance of the K-means agent (download: cancer-data.txt)
- A link to the original data set, with additional information can be found here: BreastTissue@UCI
- **DO NOT** use the original data set from the UCI link as input; I have re-formatted it to my specifications
- Include a header in the source code with the relevant information for assignments as defined in the syllabus
- Your code should **only** print the number of correctly classified testing examples followed by a newline character

- Write your report such that a peer NOT taking this course would understand the problem, your approach to solving it, justification of various choices, and your final comments
- Include plots of **all** of the statistics compiled for your report (see Example)
- Include at least one figure to illustrate the K-means method
- All sources must be properly cited; failure to do so may result in accusations of plagiarism
- Your report should be submitted in PDF format

Submission

- **Due Date: Tue. Nov. 30 by 11:00pm**
- Use your PipelineMT credentials to submit your assignment at:
  https://jupyterhub.cs.mtsu.edu/azuread/services/www/csci4350/assignment-system/public_html/

- A zipped file (.zip) containing:
  - kmeans.py
  - report.pdf

Joshua L. Phillips
Home Page | CS | MTSU

Last Modified: November 11 2021 14:41:36