INDIAN STATISTICAL INSTITUTE

Computing for Data Sciences (PGDBA)

July - December 2018

Projects

---

1. **Handwritten Digits Recognition Problem 1:**

   The MNIST data set (click here to download) is a large data set containing 70000 grayscale images of handwritten digits from 0 to 9. Each image is of size 28×28, which when flattened forms a 784 dimensional vector. Thus, each pixel forms a feature. However, all pixels do not help equally to identify the digits. The question for this problem is *How do selection or extraction of features help linear classifiers to identify handwritten digits?*. Examples of pixels that are trivial are those that are constant for all images. Among pixels that have some variation in values, not all might be needed. There are several methods for feature selection (Variance Thresholding, Select k Highest Scoring Features, Select Percentile, etc.) as well as Feature Extraction (Principal Component Analysis, Factor Analysis, Non-Negative Matrix Factorization, etc.). Investigate how the selection of features improve the accuracy of linear classification models such as Linear Regression, Linear Support Vector Machines, etc. Use a validation data set to select methods and parameters, the test data set should be used only for the final testing.

2. **Handwritten Digits Recognition Problem 2:**

   Manifold-learning is an unsupervised approach to reduce the dimensions of a data set. The data set is non-linearly transformed to a lower dimension in a manner so that points that are close to each other in the original space remain close to each other in the reduced space. The question for this problem is *How do Manifold learning methods help classifiers identify handwritten digits?* Use the MNIST handwritten digits data set, (click here to download) which is a large data set containing 70000 grayscale images of handwritten digits from 0 to 9. Each image is of size 28×28, which when flattened forms a 784 dimensional vector. Test the performance of methods such as Isomap, Locally Linear Embedding, t-distributed Stochastic Neighbor Embedding, etc. to reduce the dimension of the data. Use Kernel Support Vector Machines, Multi-Layered Perceptrons and Random Forests as the classifiers. Use a validation data set to select methods and parameters, the test data set should be used only for the final testing.

3. **Face Recognition:**

   The Extended Yale Face Database B (click here to download) contains images of 16128 images of 28 human subjects under 9 poses and 64 illumination conditions. On a number of complex problems, ensemble classifiers have been observed to work well. The question for this problem is *How well do ensemble classifiers recognize faces?*. You can consider ensemble of classifiers such as k-Nearest Neighbours as well as Random Forests of Decision Trees. Use a validation data set to select methods and parameters, the test data set should be used only for the final testing.

4. **Recognizing Song Genres:**

   The Free Music Archive Dataset for Music Analysis (click here to download) contains data sets of songs and the genres they can be classified to. You can use the features available in the fma_metadata.zip file (click here to download). Use the 16 genres classification that

is provided. Not all features might help to classify the songs. Investigate which features help to identify the genres of the songs. Train Kernel Support Vector Machines, Multi-Layered Perceptrons and Random Forests to classify the songs to their correct genres. Use a validation data set to select methods and parameters, the test data set should be used only for the final testing.

5. **Image Segmentation:**

   The BSDS 500 data set (click here to download) contains 500 color images along with groundtruth segmentation of the images. Investigate the performance of different clustering algorithms to segment the images in the BSDS 500 data set. You can use the Adjusted Rand Index to measure the accuracy of the segmentation identified by the clustering algorithm in comparison to the ground truth segmentation provided.

6. **Clustering of the 20 Newsgroup Dataset:**

   Scikit-learn contains the 20 newsgroup data set (available here) which contains news from 20 different sources. Investigate the performance of different clusteing algorithms to cluster the available news articles into the correct news groups. You can use the Adjusted Rand Index to measure the accuracy of the segmentation identified by the clustering algorithm in comparison to the ground truth labels provided.

7. **Clustering Dataset of Different Shapes:** -

   The following clustering repository (click here to visit) has a number of data sets containing clusters of different shapes and sizes. The performance of different clustering algorithms vary according to the shape and size of clusters present in the data set. Investigate the performance of different clustering algorithms on a selection of wide variety of data sets from the above repository, and try to explain why the algorithms succees or fail on different data sets. Visualizing the success and failure of the algorithms is paramount, but also you can use the Adjusted Rand Index to measure the accuracy of the segmentation identified by the clustering algorithm in comparison to the ground truth labels.

8. **Clustering by a Feature-Weighted k-Means:**

   Consider the following modification of the k-Means objective where each feature is associated with a scalar weight:

$$\min_{w,V,U} J_{WKM} \sum_{j=1}^{k} \sum_{i=1}^{n} \mu_{ij} ||x_i - v_j||^2$$

$$\text{subject to, } \mu_{ij} \in \{0,1\}, \sum_{j=1}^{c} \mu_{ij} = 1.$$

   Here the data set is $X = \{x_1, ..., x_n\}, x_i \in R^d$, the set of cluster memberships $U = [\mu_{ij}]_{(n \times d)}$ contains in each row the membership of each data point to every cluster, the set of cluster centers is $V = \{v_1, ..., v_j\}, v_i \in R^d$, and the set of feature weights is $w = \{w_1, ..., w_d\}$. The i-th feature is associated with a scalar weight w_i that indicates how important the feature is.

   Derive update expressions for w as well as all v_i and μ_{ij}. Implement the update expressions in an Alternating Optimization algorithm, and test its performances across different datasets from the following repository clustering benchmark. Compare the performance with that of k-Means. Additional questions: Does the Alternating Optimization algorithm converge? What is its time complexity?

9. **Object Classification by Finetuning Deep Convolution Networks:**

   The objective of this problem is to train a Deep Convolution Network to classify the CIFAR 10 and CIFAR 100 data sets. Take a pre-trained model (e.g., VGG-19) and replace the last layer so that it contains the proper number of classes for the data set at hand. Perform two

types of finetuning of this modified network (i) Train the entire network on the data set (ii) Train only the weights between the penultimate and last layer. Compare the performances of the two networks.

10. **Object Classification by Extracting Features from Deep Convolution Networks:**

    The objective of this problem is to perform object classification on the CIFAR 10 and CIFAR 100 data sets, by extracting features from a fine tuned (see project 9) pre-trained Deep Convolution Network (e.g., VGG-19). Consider the output of the penultimate layer as features. Use the extracted features to train a Kernel Support Vector Machine and a Random Forest classifier. Compare the performances of the two learned classifiers on the two data sets.