Assignment 1

Imran Salim

April 27, 2019

1 Introduction

The spherical k-means clustering algorithm is used to find the most accurate positions of a set of clusters for a provided dataset. This is done in order to construct a dictionary of filters. The clustering continues to run for n iterations until convergence is achieved. After convergence is achieved, you will be left with a dictionary of filters which can be used to learn representations for the purpose of accurately classifying images.

The goal of this assignment was to apply the unsupervised learning algorithm, spherical k-means [1], to extract features from images divided into subsections. A dictionary is used to store the features which can then act as filters. The filters are run over images in an attempt to classify an image correctly.
The WEKA workbench [2] was used to implement this machine learning algorithm in the Java programming language.

The data that was provided to these algorithms were the:

- MNIST dataset of handwritten numbers,
- Fashion-MNIST dataset of fashion products,
- SVHN dataset of house numbers in their respective streets,
- CIFAR-10 dataset of animals and vehicles.

2 Method

The spherical k-means algorithm is used to apply a feature learning technique for the given datasets in this assignment.
During the pre-processing stage, the input is split into sub-sections called patches. These patches are then normalized. A whitening or sphering transformation is applied to the normalized patches in order to remove correlations by scaling the values to unit length i.e. linear relationships between variables are removed. This is the step that distinguishes the spherical k-means from the regular k-means.

Now the spherical k-means algorithm can start working on the data.
A dictionary is initialized with points randomly selected over a normal distribution. These random variables are then scaled to unit length via whitening. In practice, the dictionary is represented as a matrix with the number of patches by the number of centroids i.e. the filters. Calculating this matrix requires the help of two other matrices. One which is full of the whitened data, and another which has to be calculated. The matrix that has to be calculated contains a row for every centroid, and for each of these rows, there are columns corresponding to the the number of patches. Each of these columns will have one non-zero value in them which represents how closely related the patch is to a centroid i.e. a row in the matrix.
The dictionary is updated by calculating the dot product of the two matrices. The values in the dictionary have to be whitened for every update iteration. This process of iterative re-positioning of the clusters is repeated until convergence is achieved i.e. the sum of squared errors between the previous values of the centroids and the new ones calculated in the update is infinitesimal. The result is a feature bank which can be used as filters to classify images.
During the process of classification, the image is divided into patches once again. A filter slides across the image and the cross-correlation is calculated between the filter and the corresponding patch. The results of these calculations are your feature maps. The feature maps are fed into a non-linear activation function which removes negative values by setting them to 0. A process called average pooling is then applied to the feature maps. Average pooling is where the average of a square subsection of pixels in the feature map is calculated. The purpose of this is to reduce the resolution of the feature map while retaining the required information used for classification, thus, less resources are required.
The final result is a feature vector which describes an instance. These instances are used to aid the learning process when the model is being trained.

3 Experimental results

Some results are shown in Table 1. These were generated using WEKA.

	MNIST	Fashion-MNIST	SVHN	CIFAR-10
Correctly classified instances on training data	99.89%	95.88%	92.88%	78.57%
Incorrectly classified instances on training data	0.11%	4.12%	7.12%	21.43%
Root mean squared error of the training data	0.2716	0.2723	0.2728	0.2749
Correctly classified instances on testing data	98.4%	89.07%	86.07%	64.13%
Incorrectly classified instances on training data	1.6%	10.93%	13.93%	35.87%
Root mean squared error of the testing data	0.2718	0.2732	0.2744	0.2783

Table 1: Results from applying this feature learning technique to four datasets.

From these results, you can see that the more complicated each dataset gets, the more difficult the task of classification becomes. Factors that complicate the data are whether they contain a wide variability in color range of each pixel. MNIST and Fashion-MNIST contains pixels that are grayscale and the instances belonging to their respective class are uniform. SVHN and CIFAR-10 are not grayscale which introduces a lot more variability in the pixel values of the images. CIFAR-10 has the widest variability in pixel values and uniformity which may be why that it has the lowest ratio of correct and incorrect classifications.

4 Conclusions

The spherical k-means clustering algorithm, a variant of the k-means clustering algorithm, can be used in an attempt to train a model such that it will accurately classify a set of instances in a dataset.
This can be done very well in a controlled environment i.e. a carefully constructed dataset to be used solely for the purpose of classification. The reliability of the model drops significantly when it is trained on data with a lot of confounding factors. Data of this nature is more like the kind that will be encountered when trying to solve problems of a much larger scope.
Problems of this magnitude, if solved, would be greatly beneficial to society.

References

[1] A. Coates and A. Y. Ng. Learning feature representations with k-means. In *Neural Networks: Tricks of the trade*, pages (561–580). Springer, 2012.

[2] Ian H. Witten, Eibe Frank, and Mark A. Hall. *Data Mining: Practical Machine Learning Tools and Techniques.* Morgan Kaufmann, Burlington, MA, 3 edition, 2011.