Your Name
email@fas.harvard.edu
CS181-S16

Assignment #4
Due: 5:00pm April 1, 2016

Collaborators: John Doe, Fred Doe

Homework 4: Clustering

There is a mathematical component and a programming component to this homework. Please submit ONLY your PDF to Canvas, and push all of your work to your Github repository. If a question requires you to make any plots, please include those in the writeup.

Problem 1 (The Curse of Dimensionality , 5pts)
To be released!

Solution

Problem 2 (The Curse of Dimensionality, 5 pts)
To be released!

Solution

K-Means [15 pts]

Implement K-Means clustering from scratch.[1]. You have been provided with the MNIST dataset. You can learn more about it at http://yann.lecun.com/exdb/mnist/. The MNIST task is widely used in supervised learning, and modern algorithms with neural networks do very well on this task. We can also use MNIST for interesting unsupervised tasks. You are given representations of 6000 MNIST images, each of which are 28x28 handwritten digits. In this problem, you will implement K-means clustering on MNIST, to show how this relatively simple algorithm can cluster similar-looking images together quite well.

Problem 3 (K-means, 15pts)

The given code loads the images into your environment as a 6000x28x28 array. Implement K-means clustering on it for a few different values of K, and show results from the fit. Show the mean images for each class, and select a few representative images for each class. You should explain how you selected these representative images. To render an image, use the numpy imshow function, which the distribution code gives an example of. Use squared norm as your distance metric. You should feel free to explore other metrics along with squared norm if you are interested in seeing the effects of using those. Also, your code should use the entire provided 6000-image dataset (which, by the way, is only 10% of the full MNIST set).

Are the results wildly different for different restarts and/or different K? Plot the K-means objective function as a function of iteration and verify that it never increases.

Finally, implement K-means++ and see if gives you more satisfying initializations (and final results) for K-means. Explain your findings.

As in past problem sets, please include your plots in this document. You may have many plots for this problem, so feel free to take up multiple pages, as long as it is organized.

Solution

[1] That is, don't use a third-party machine learning implementation like `scikit-learn`; `numpy` is fine.

Problem 4 (Calibration, 1pt)
Approximately how long did this homework take you to complete?