Data Mining: Learning from Large Data Sets - Spring Semester 2014
Instructor: Prof. Andreas Krause

Course Project
Part III

Extracting Representative Elements From Large Datasets

1 Introduction

The goal of this task is to extract representative elements from a large image dataset. We will define the representativeness of a set of elements C for a given set P is defined as:

$$\frac{1}{|P|}\sum_{p\in P}\min_{c\in C} d(p,c)^2. \tag{1}$$

where $d(p,c)$ is the Euclidean distance between vectors p and c. Optimal C would be P which would minimize expression 1. However, we are limited to picking exactly k elements from C. One way to approach this problem is to apply K-$Means$ clustering on P. The representative elements would then be the points from the original dataset closest to the computed k centers.

Finding C of cardinality k that minimizes expression 1 is NP hard. Commonly used heuristic is the classic Lloyd's algorithm with a smart initialization strategy called K-means++. The computational complexity of the classic Lloyd's algorithm is prohibitive when it comes to solving larger problem instances.

In order to tackle this problem we will consider algorithms that fit in the MapReduce computational model.

2 Dataset Description

You will be given a subset of the Tiny Images [1] which you can download from here: `http://las.ethz.ch/courses/datamining-s14/datasets/tiny_subset`. In the original representation, each image was an integer vector of dimension 3072 ($32\times 32\times 3$, intensity was given for each pixel). We have performed mean normalization, feature scaling, dimensionality reduction with PCA, as well as whitening. We then extracted a subset from that dataset which contains 100,000 images, each being a 750 dimensional feature vector. Further feature transformations are not allowed. The dataset has been serialized to the "npz" format which enables efficient loading. File "data.py" shows how to load the dataset. **Note that your mappers will receive vectors as space delimited floats, one vector per line (as in previous tasks).** The full training set on the cluster is 2,000,000.

3 Evaluation and Grading

Your task is to provide a Map function and a Reduce function written in Python.

The output of the Reduce function should contain $k = 200$ lines. Each line should contain the 750 dimensional vector representing one of the centers. Note that the centers don't have to be elements of the training dataset.

[1] http://horatio.cs.nyu.edu/mit/tiny/data/index.html

Your model will be trained on a larger subset of the Tiny Images dataset. The centers outputed by the reducer will be used to compute the score of your submission on a separate dataset:

$$cost(P, C) = \frac{1}{|P|} \sum_{p \in P} \min_{c \in C} d(p, c)^2. \tag{2}$$

We will compare the score of the submission with two baseline predictions: a weak one (called "baseline easy") and a strong one (called "baseline hard"). These will have a score of PBE and PBH respectively, calculated as described above. Both baselines will appear in the rankings together with the score of your submitted predictions.

Performing better (having lower score) than the weak baseline will give you 50% of the grade, and matching or exceeding the strong baseline on the **test set** will give you 100% of the grade. This allows you to check if you are getting at least 50% of the grade by looking at the ranking. If your prediction performance on the **test set** ($\mathsf{P}_{\mathsf{test}}$) is in between the baselines, the grade is computed as:

$$\text{Grade} = \left(1 - \frac{\mathsf{P}_{\mathsf{test}} - \mathsf{PBH}_{\mathsf{test}}}{\mathsf{PBE}_{\mathsf{test}} - \mathsf{PBH}_{\mathsf{test}}}\right) \times 50\% + 50\%$$

The number of submissions per team is limited to 50. Time limit per each mapper and reducer is 15 minutes. There is also a memory limit of 2G. The number of mappers that will be used is 300. There will be only one reducer. The submission with the lowest score (best grade) on the test set will be used for grading.

3.1 Report

At the end of the course you will be requested to provide a team report which describes your solution for all project tasks. You should update the template given with the first task and add a section titled "Extracting Representative Elements From Large Datasets". The maximum length of a task report is 2 pages (which means that the final report will contain a maximum of 8 pages). You will be required to upload the reports **after the last project of the course.**

3.2 Deadline

The submission system will be open from **Tuesday, 22.04.2014, 17:00** until **Tuesday, 13.05.2014, 23:59:59**.