Project
Data Mining: Learning from Large Data Sets
FS 2015

Project 3, Nov 19, 2015
Extracting Representative Elements

LAS Group, Machine Learning Institute
Dept. of Computer Science, ETH Zürich
Prof. Dr. Andreas Krause
Web: `las.ethz.ch/courses/datamining-f15/`
Email questions to:
Alkis Gotovos `<alkisg@inf.ethz.ch>`
Josip Djolonga `<josipd@inf.ethz.ch>`

1 Introduction

The goal of this project is to extract representative elements from a large image data set. If we extract a representative subset $\mathcal{C}$ from a data set $\mathcal{D}$, the quality of our selection is quantified by the average distance of the elements of $\mathcal{D}$ to their closest representative in $\mathcal{C}$, that is,

$$Q(\mathcal{C}) = \frac{1}{|\mathcal{D}|} \sum_{x \in \mathcal{D}} \min_{c \in \mathcal{C}} \|x - c\|^2. \tag{1}$$

Ideally, we would like to minimize the above expression under the constraint that we pick k elements to include in $\mathcal{C}$, which is equivalent to performing K-means clustering. However, this is in general an NP-hard problem, so we have to resort to heuristic methods, such as Lloyd's algorithm. For large data sets, even Lloyd's algorithm can be computationally prohibitive, therefore in this project you will solve this problem using the map-reduce framework.

2 Input and Output Specification

You are given a training set in the form of a text file `train.txt` consisting of 100,000 lines. Each line corresponds to one image, and contains 500 features (space-separated floats) that we extracted from the original data set. You are asked to implement an algorithm that computes representative elements in feature space using map-reduce. As in previous projects, you need to provide two Python files, a mapper and a reducer, templates of which are provided together in this handout. We are also providing a script to locally check the quality of your solution according to (1).

Your reducer should output $k = 100$ lines of representative elements, each line containing 500 space-separated floats. Note that the elements you output do not have to be elements of the input data set. Your submissions will be evaluated on a much larger data set on the server. There is a time limit of 15 minutes per submission on the server. You are allowed 25 submissions per team, and the one with the highest score will be used to determine your grade.

3 Evaluation and Grading

The solution of your submission is evaluated on the server test set according to (1). Note that the score is evaluated on a different set of data than the training one. We will compare the Q score of your submission to two baseline solutions: a weak one (called "baseline easy") and a strong one (called "baseline hard"). We denote the scores of these baselines QBE and QBH respectively. Both baselines will appear in the rankings together with the Q score of your solutions. Performing better 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 Q score lies between the baselines, your grade is computed as follows:

$$G = \left(1 - \frac{Q - \mathrm{QBH}}{\mathrm{QBE} - \mathrm{QBH}}\right) \times 50\% + 50\%.$$

3.1 Report

You have to submit one report per group describing your work. Please keep the reports brief (under 2 pages). You are requested to upload a ZIP archive containing the team report *and* the code. We include a template for LaTeX in `report.tex`. The reports are uploaded on the same page as the test set submissions.

3.2 Deadline

The submission system will be open from **Thursday, 19.11.2015, 17:00** until **Wednesday, 02.12.2015, 23:59:59**.