Project
Data Mining: Learning from Large Data Sets
FS 2015

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:
lucic@inf.ethz.ch

Project 1, Oct 1, 2015
Locality Sensitive Hashing

1 Introduction

In this project we are interested in detecting duplicate videos. Suppose that there exists a company which offers a web service that allows users to upload various (cat) videos. When an user uploads a copyrighted video, the company is sooner or later sued by the copyright owner. Your task is to develop an efficient method to to detect whether a video is a near-duplicate of a copyrighted video, in which case you don't want to allow the user to publish it.

2 Input and Output Specification

You are given two text files: "training.txt" and "duplicates.txt". Each line of the training dataset contains the features for one video file and is formatted as follows: string "VIDEO XXXXXXXXX" followed by a list of space delimited integers in range $[0, 20000]$. You can consider them equivalent to shingles in the context of near-duplicate document retrieval. We will use Jaccard similarity based on the video features as the similarity metric. The "duplicates file" contains the pairs of near duplicates (videos that are at least 90% similar according to the Jaccard similarity). Each line is a tab-delimited pair of integers where the first integer is always smaller.

You are asked to develop a Locality Sensitive Hashing solution implemented using map-reduce. You need to provide two files: implementation of the mapper and reducer in Python. You can find the template code for both the mapper and reducer in this handout. Furthermore, there is a script that you can use to check the F_1 score of your submission on the small training dataset. You are free to change the mapper and reducer in any way you see fit, as long as the following holds:

- The maximum number of hash functions used is 1024.
- The maximum total running time of the reducer is 15 minutes.
- Mapper should read lines from standard input (formatted as in "training.txt").
- Reducer outputs a pair of integers representing the ID's of duplicated videos on separate lines (identical format as in the "duplicates.txt").
- Output each pair only once.
- Use only the following extra packages: NumPy, SciPy

Each submission on the project website will be run against a large scale dataset, but the sample given to you is representative, in a sense that you can tune the parameters of your algorithm locally (up to a certain degree). The number of submissions per team is limited to 25. The submission with the highest F_1 score will be graded.

3 Evaluation and Grading

For each line of the output of your algorithm we check whether the reported videos are in fact at least 90% similar. If they are, this row will count as one true positive. If they are not, it will count as one false positive. In addition, each video pair that is similar but was not reported by your algorithm will count as one false negative. Given the number of true positives, false positives and false negatives, TP, FP and FN respectively, we can calculate:

1. **precision**, also referred to as Positive predictive value (PPV), as $P = \frac{TP}{TP+FP}$.
2. **recall**, also referred to as the True Positive Rate or Sensitivity, as $R = \frac{TP}{TP+FN}$.

Given precision and recall we will calculate the F_1 score defined as $F_1 = 2PR$. We will compare the F_1 score of your submission to two baseline solutions: a weak one (called "baseline easy") and a strong one (called "baseline hard"). These will have the F_1 of FBE and FBH respectively, calculated as described above. Both baselines will appear in the rankings together with the F_1 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 F_1 score is in between the baselines, the grade is computed as:

$$G = \left(1 - \frac{FBH - F_1}{FBH - FBE}\right) \cdot 50\% + 50\%.$$

3.1 Report

You are requested to upload a ZIP archive containing the team report *and* the code. We included a template for LaTeX in the file `report.tex`. Please keep the reports brief (under 2 pages). If you do not want to use LaTeX, please use the same sections as shown in `report.pdf`. The reports are uploaded on the same page as the test set submissions.

3.2 Deadline

The submission system will be open from **Thursday, 01.10.2015, 17:00** until **Wednesday, 14.10.2015, 23:59:59**.