Fast Nearest Neighbor Search Implementation

Team Members:

John Curci – jcurci92@bu.edu

Fei Chen – cf0906@bu.edu

Shuang Zhao – zs1995@bu.edu

Project Motivation

* Nearest Neighbor Search is a machine learning technique that relies on using some distance metric to search for vectors that are close to each other in high dimensional space. It solves the general problem of finding “similar” results among complex items such as images, documents, etc.
* A naive implementation of Nearest Neighbor Search requires comparison of each test point to the entire set of training data. This means that search runs in O(**n\*d**) time with the size of the training set **n** and the number of features **d**. This results in a prohibitively slow search for large sets of complex items. A practical approach requires a reduction in both the number of comparisons and the complexity of each comparison.
* The goal of this project is to implement two techniques for approximating the NN search algorithm which reduce the number of comparisons (**n**) required to find the k-nearest neighbors of a given test vector, and then to apply this to create a search algorithm over a training dataset.

Requirements

1. Implement a k-dimensional tree (kd-tree) approach to Nearest Neighbor Search. This approach reduces the number of data points compared by iteratively dividing the training data in half, and only comparing points which are in the same half for every division, which are located at search time by navigating through a binary tree.
2. Implement a locality sensitive hashing approach to Nearest Neighbor Search as described in [1]. This approach uses a hash function to place all training data points into fixed-size groups, and only compares each test point to training points in the same bucket. This approach produces an approximate result, because closely located points are likely, but not guaranteed, to be placed in the same group.
3. Optional – deploy both search algorithms to a web application where users can upload images and display the k-nearest neighbors from the dataset.

Dataset

We are using the “80 million tiny images” dataset from MIT, as recommended in the sample project list. This data has already been significantly processed, which will reduce the implementation complexity of our code. The challenge of working with this dataset is that it is very large, over 200GB for the raw images and over 100GB for the gist data. This will require storing the data and in a database and querying it in batches.

Goals for Midterm Report

1. Read literature on LSH/k-d tree NN approximation
2. Determine approach (e.g. coding language, platforms needed, etc.)
3. Deploy dataset to storage/operation environment (e.g. cloud database)
4. Begin implementation of prototype algorithms to be tested on small synthetic datasets

References

[1] A. Rajaraman & J. Ullman (2010). ["Mining of Massive Datasets, Ch. 3"](http://infolab.stanford.edu/~ullman/mmds.html)