Mining Massive Data

Programming Assignment 1

Locality Sensitive Hashing

Zoheir El Houari

Brief discussion about the implementation of LSH and the approximate nearest neighbour algorithm

Our implementation of the LSH class was based on the code provided in the blog post (see references). The HashTable class was updated and modified in order to sample the random projection matrix using Achlioptas's method using the choice function from the random library, which then samples from an input values with the specified weights.

The constructor of the LSH class generates an empty results table and a list of empty **HashTable** classe. The method **calculate_hashes** projects the feature vectors into a lower dimensional space and converts them to a binary format, i.e. it fills the empty hash tables ofthe LSH class with the computed signatures of each feature vector.

finding_similarities uses the method_**getitem** in order to union all the track IDS in the hash tables that corresponds to have the same signature as the track IDs in the validation or test subsets.

calculating_similarities calculates both the cosine and Euclidian similarities between each validation track_id and the similar tracks predicted by the LSH model.

k_nearest_neighbor

This function wasn't finished. So far in the implemented version we were able to get the sort list of similar tracks based on their similarity.

Unfortunately we did not manage to finish the prediction part for this function. But the idea was to slice the resulting sorted lists to get the exact k neighbours and to perform counts foreach selected genre, the genre with the highest count would be considered as the predicted genre for the track at hand.

The function classification_accuracy()

is aiming to perform the assessment of the accuracy of the above mentioned algorithm, but the algorithm wasn’t fully implemented we didn’t have a chance to properly test it. The idea was to count the percentage of correct predictions for both selected measures (euclidean distance and cosine similarity).

Detail how you trained your algorithm and how you performed the hyperparameter optimization. Report tested parameter and the results for these parameters.

To get an idea about how different hyperparameter values will affect the accuracy of our algorithm, we thought about taking the brute force approach, but due to the lack of contribution of a team member, we couldn’t manage to do this task.

Detail why you settled on a specific choice of l (hash length), n (number of hash tables), k (number of nearest neighbours for the prediction) and similaritymeasure m

The choice of the aforementioned parameters (i.e. l, n and k) was conducted through empirical experiment. In other words the choice was based on the numerous guesses, and following tests.

In regard to the similarity measures we tested both metrics recommended in the assignment description, namely cosine similarity and euclidean distance. But unfortunately as our k nearest neighbours algorithm wasn’t completed we didn’t manage to properly test the implemented measures.

Report the classification accuracy of your algorithm on the test set.

The classification accuracy function implemented by our team was supposed to calculate the portion of correctly classificated genres by our algorithm based on both measures. The idea was to use during e[periments to figure out which measure and what parameter set would perform the best in the given case.

Comment on why the chosen random projection method could be beneficial todrawing rij from a Gaussian distribution?

While a projection of points onto a spherically random hyperplane through the origin is simple, it’s also a nontrivial task, especially when dealing with databases. As a result of therandom projection method (Achlioptas method) it was introduced as a more efficient and simple arithmetics. The advantage of the random projection method is that it achieves the same embedding while being time efficient which leads to a much less time spent on computation for reducing the dimensions.

How much time did you spend on the assignment (including the writing of thereport; please provide an average for all of your team members)? This will notbe used for grading.

On average each member of the team spent approximately 30 hours working on this assignment.

References:

https://towardsdatascience.com/locality-sensitive-hashing-for-music-search-f2f1940ace23