COMS E6998-9: Algorithmic Techniques for Massive Data Nov 6th, 2015

Project Proposal - LSH w/o False Negatives

By: *Yang, Shekel Nosatzki, Loganathan (UNI:* cy2417, ns3049, pl2487*)*

1 Background

As we have seen in class, most LSH algorithms are using random hash functions, each with high probability of collision of close points, and low probability of collision of far points, and taking many of such hash tables, each *independent* from one another, we can prove that with very high probability close points will collide and far points will not collide. The subject article provide a fresh look into the problem by taking hash functions that are *dependent* of each other in a way, that removes the low probability that close points will never collide.

The article focuses on the Hamming Space, where a hash function is coordinate sampling. Instead of sampling random coordinates on each hash function, the author suggests a method to pre-select sampling that are *"r-covering"*, using a generating function m. The author shows that under certain assumptions on the n,r,c parameters, the algorithm can match the performance of non-deterministic algorithms which do not guarantee success.

2 Extension of Findings

In our project, we will attempt to extend the idea in the article. We will explore several directions:

- **Other metric spaces** - since non-trivial embeddings have randomization, our focus will be in implementing correlated hash functions in l_1, l_2, and possibly other metric spaces. We will consider looking at simplified discreet versions of such spaces (e.g. integers only).

- **Data-Dependent Hashing** - the article assumes no a-priori information about our data. Here we can look not only on having the hash functions dependent on each other, but also on the data, to make sure collision are found. We will focus on a *"random-case"* - i.e. where the data is behaving "as expected", and will consider different metrics for such an extension.

- **Minhash** - We will consider the effect of the hash function construction in the case of sparse vectors and suggest improvements to the algorithm if such are necessary.

- **Different Settings** - While the article does generalize the performance under any settings (mainly in terms of n, r, c) - the focus of the article is on certain setting which yield $\rho = \frac{1}{c}$. We will consider different settings which generate sub-optimal performance, and see if there is any room for improvement under such settings.

Depending on our success in finding a concrete extension - we will either focus on that concrete extension or analyze the observations we will find in different extension attempts.