Review on NLP based Technique to Improve the Performance of knn

Tejaswini Channe, Gurudev Sawarkar
Department of Computer Science & Engineering, Nagpur University
W.C.E.M. Dongargaon, Nagpur, India

Abstract—Conventional techniques such as spatial inverted index used for nearest neighbor search extends the conventional inverted index to cope with multidimensional data, and also able to answer the nearest neighbor queries with keywords in real time. For example, instead of considering all the restaurants, a nearest neighbor query would instead ask for the restaurant that is the closest among those whose menus contain “steak, spaghetti, brandy” all at the same time. But this technique has a few deficiencies that seriously impact its performance. Motivated by this, I develop a newly access method based on natural language processing that improves the performance of k nearest neighbor search.

Keywords—NLP, Nearest Neighbour Search, Keyword Search, knn, SPI.

I. INTRODUCTION

Traditional techniques, such as spatial inverted index and nearest neighbour retrieval, find objects satisfying both a spatial predicate, and a predicate on their associated texts. Spatial inverted index is optimized for multidimensional points. It incorporates point coordinates into a conventional inverted index with small extra space, owing to a delicate compact storage scheme. There are easy ways to support queries that combine spatial and text features.

A spatial database manages multidimensional objects (such as points, rectangles, etc.), and provides fast access to those objects based on different selection criteria. The importance of spatial databases is reflected by the convenience of modelling entities of reality in a geometric manner. For example, locations of restaurants, hotels, hospitals and so on are often represented as points in a map, while larger extents such as parks, lakes, and landscapes often as a combination of rectangles. Many functionalities of a spatial database are useful in various ways in specific contexts. For instance, in a geography information system, range search can be deployed to find all restaurants in a certain area, while nearest neighbour retrieval can discover the restaurant closest to a given address.

Today, the widespread use of search engines has made it realistic to write spatial queries in a brand new way. Conventionally, queries focus on objects’ geometric properties only, such as whether a point is in a rectangle, or how close two points are from each other. We have seen some modern applications that call for the ability to select objects based on both of their geometric coordinates and their associated texts. For example, it would be fairly useful if a search engine can be used to find the nearest restaurant that offers “steak, spaghetti, and brandy” all at the same time. Note that this is not the “globally” nearest restaurant (which would have been returned by a traditional nearest neighbor query), but the nearest restaurant among only those providing all the demanded foods and drinks.

There are easy ways to support queries that combine spatial and text features. For example, for the above query, we could first fetch all the restaurants whose menus contain the set of keywords {steak, spaghetti, brandy}, and then from the retrieved restaurants, find the nearest one. Similarly, one could also do it reversely by targeting first the spatial conditions browse all the restaurants in ascending order of their distances to the query point until encountering one whose menu has all the keywords. The major drawback of these straightforward approaches is that they will fail to provide real time answers on difficult inputs. A typical example is that the real nearest neighbor lies quite far away from the query point, while all the closer neighbors are missing at least one of the query keywords.

But there are many disadvantages of these straightforward approaches. So I develop a new technique which is based on NLP to improve the performance of nearest neighbour search. In this paper I propose a new technique which is based on NLP that improves the performance of k nearest neighbor search.

II. LITERATURE SURVEY

In the paper ‘fast nearest neighbor search with keywords’, there are methods like spatial index, inverted index, nearest neighbor search. The first method spatial index is used for creating indices because there is huge amount of data need to be stored for searching that data stored in the form of xml documents. If the data storage created in the form of indices then space required is less also time needed for searching the keyword is less.
Second method is inverted index. The inverted index data structure is a central component of a typical search engine indexing algorithm. A goal of a search engine performance is to optimize the speed of the query: find the documents where word occurs. Once an index is developed, which provisions lists of words per document; it is next inverted to develop an inverted index. Querying the index would require sequential iteration through each document and to each word to verify a matching document. The time memory and processing property to execute such a query are not always theoretically realistic. Instead of listing the words per article in the index, the inverted index data structure is developed which lists the documents per word. The inverted index produced, the query can now be determined by jumping to the word id in the inverted index. These were effectively inverted indexes with a small amount of supplementary explanation that required a implausible amount of attempt to produce. The spatial inverted list (SI-index) is essentially a compressed version of an I-index with embedded coordinates. Compression is already widely used to reduce the size of entire document an inverted index in the conventional context where each inverted list contains only ids. In that case, an effective approach is to record the gaps between consecutive ids, as opposed to the precise ids. For example, given a set S of integers {2, 3, 6, 8}, the gap-keeping approach will store {2, 1, 3, 2} instead, where the i-th value (i ≥2) is the difference between the i-th and (i − 1)-th values in the original S. As the original S can be precisely reconstructed, no information is lost. The only overhead is that decompression incurs extra computation cost, but such cost is negligible compared to the overhead of I/Os. Note that gap-keeping will be much less beneficial if the integers of S are not in a sorted order. This is because the space saving comes from the hope that gaps would be much smaller (than the original values) and hence could be represented with fewer bits. This would not be true had S not been sorted. Compressing an SI-index is less straightforward.

Third method is nearest neighbor search. Nearest neighbour search (NNS), also identified as closeness search, parallel search is an optimization problem for finding closest points in metric spaces. In the paper “Efficient Keyword-Based Search for Top-K Cells in Text Cube” methods used are inverted-index one-scan, document sorted-scan, bottom-up dynamic programming, and search-space ordering. In the top k cells, there is a searching of nearest key to the query. Cubes forms clusters of single unique group which shows its identity. Method like inverted index used for giving index rather than providing whole data which can be space consuming.

### III. IMPROVED TECHNIQUE FOR KNN

This paper is focused on to improve the performance of knn with the help of natural language processing. As we know knn is one of those algorithm that are very simple to understand but work incredibly well in practice. Knn is non parametric lazy learning algorithm, it does not use training data points to do any generalization that is there is no explicit training phase but costly testing phase cost in terms of time and memory. With the help of proposed technique cost will be minimized.

#### A. Knn algorithm

K-nearest neighbor algorithm (KNN) is part of supervised learning that has been used in many applications in the field of data mining, statistical pattern recognition and many others. KNN is a method for classifying objects based on closest training examples in the feature space. An object is classified by a majority vote of its neighbors. K is always a positive integer. The neighbors are taken from a set of objects for which the correct classification is known.

The algorithm on how to compute the K-nearest neighbors is as follows:

1. Determine the parameter $K = \text{number of nearest neighbors}$ beforehand. This value is all up to you.
2. Calculate the distance between the query-instance and all the training samples. You can use any distance algorithm.
3. Sort the distances for all the training samples and determine the nearest neighbor based on the $K$-th minimum distance.
4. Since this is supervised learning, get all the Categories of your training data for the sorted value which fall under $K$.
5. Use the majority of nearest neighbors as the prediction value.

But this algorithm has some drawback such as follows: There is need to determine value of parameter $K$ (number of nearest neighbors) Distance based learning is not clear which type of distance to use and which attribute to use to produce the best results. Shall we use all attributes or certain attributes only? Computation cost is quite high because we need to compute distance of each query instance to all training samples.

#### B. NLP based technique

Natural Language Processing enables communication between people and computers and automatic translation to enable people to interact easily with others around the world. Natural language processing is a field of computer science, artificial intelligence, and linguistics concerned with the interactions between computers and natural languages.

The proposed technique is based on NLP in which keyword extraction is done automatically from the text document, there is no need to create separate database to save important words from the text document. For automatic extraction of words from text document so that it overcomes the drawback of knn and will improve the performance.

### IV. CONCLUSIONS

From the study I conclude that the technique based on natural language processing for performance improvement of knn will give better outputs than the traditional techniques.
REFERENCES


