Parth Shah

Neurocomputing

Complete Summary of Article:

Efficient kNN classification algorithm for big data Zhenyun Deng, Xiaoshu Zhu n , Debo Cheng, Ming Zong, Shichao Zhang n Guangxi Key Lab of Multi-Source Information Mining & Security,Guangxi Normal University, Guilin, Guangxi 541004, China

Purpose:

The main purpose of this article is to find the best kNN classification method to deal with big data. They are using **new kNN technique** instead of **traditional kNN** method because it has very high liner time complexity on the sample data sets, thus it is **not** work well on **big data set**. The new kNN algorithm uses the new training process, which blocks the dataset by clustering algorithm, and during the testing process it classify the test samples within a cluster according to similarity of that cluster i.e nearest cluster has high similarity and low similarity between the clusters. Moreover, in this new kNN algorithm they have used **Ladmark-based Spectral Clustering** instead of previous clustering method (like grid-based clustering, and density clustering) because this clustering algorithm scales the data in linear way, and it has low complexity.

How to pick best m and k values:

Now, they need to **solve some of the questions** in order to pick best kNN classification. Those questions can be how to select best **landmark algorithm**, **accurate m, and k values etc. Firstly**, to select the landmark, they have two very popular and effective methods 1) random sampling and 2) k-mean based method. In this paper they have **used k-means algorithm.** This algorithm **significantly reduces** the time complexity (from $O(n^3)$ to $O(n)$) when they compute all 4 equations. **Secondly**, when they apply LC-kNN algorithm then the question is how to set the suitable value of **m** (m is number of clusters which are the output of the k-means algorithm) because when the m is increasing, then at the same time accuracy is decreasing, and if m is too small still it leads to low accuracy when dealing with big data. Thus, to set m in a suitable range. You need to assume that the proposed algorithm needs M memory space, for example, then the memory of PC should be M1 along with the smallest class of training sample is n0. Hence, the value range of m can be expressed by $M/M_1 < m < n_0$.

The main thing is to find the perfect value of m, that's why they **conducted** many experiments by choosing different values of m ranging from 10 to 30, and made a comparison between kNN, LC-kNN, and RC-kNN. At the end they found that LC-kNN and RC-kNN algorithms perform very well as compared to kNN. It turns out that both algorithms were closer to kNN classification accuracy with m=10, thus they have decided to choose m=10 (**successfully found the accurate value of m**). **Lastly**, Now the **next challenge** is to **find the k** (number of nearest neighbors or clusters) value. They have tried different k values ranging from 0 to 20, but according to experiment when the k value is increasing then the accuracy is continuously decreasing, thus for them the perfect value of k is 1 because when the **k =1**, the performance/accuracy of LC-kNN is higher.

Summary of all the experiments:

Finally, after all experiments they observed that in terms of **time complexity** the **RC-kNN and LC-kNN performed very well as compared to kNN**, though **accuracy and time is much better in LC-kNN than the RC-kNN.**

Summary:

In conclusion, the result shows that the proposed kNN classification worked performing very well in terms of accuracy and efficiency in the different data sets (like USPS, MNIST, LETTER, etc.) model. However, if we compare the data then LC-kNN works better than proposed kNN because it has very less time complexity, hence, according to me the **data and conclusion does not match with each other**.

Some key things to note:

Following are **some key things** that **I learned** from this article:

- ➢ How new kNN classification **work perfectly on big data** using the LSC (Landmark-based Spectral Clustering) clustering algorithm.
- ➢ Choosing the **right value of m** is very important as it is directly affected on model's performance (not too small and too big). We can try using the multiple different values (like 10, 15, 20, etc.) then decide what is best value according to accuracy and time.
- ➢ The **k (cluster) value** should be set as **small** as possible in the case of high classification accuracy (sometimes k=1 can be fine).
- ➢ **We can compare different algorithms** (like LC-kNN, and RC-kNN) and pick **the best one** that works well in your model. Which satisfy both higher accuracy and lower time.