International Journal of Contemporary Research in Computer Science and Technology (IJCRCST) ISSN: 2395-5325
Volume 3, Special Issue 1 (February '2017)
Special Issue on 1st National Conference on Advanced Computing and Communication Technologies,
Conference held at Siri PSG Arts and Science College for Women, Sankari,Tamilnadu,India.

EFFICIENT KNN CLASSIFICATION ALGORITHM FOR BIG DATA

G.Priya,
Research Scholar,
Department of Computer Science,
Theivanai Ammal College for Women,
Villupuram,Tamilnadu,India.

K.Manohari,
Assistant Professor
Department of Computer Science,
Theivanai Ammal College for Women,
Villupuram,Tamilnadu,India.

Abstract: K Nearest Neighbors (kNN) is an efficient lazy learning algorithm and has successfully been developed in real applications. It is natural to scale the kNN method to the large scale datasets. In this paper, we propose to first conduct a k-means clustering to separate the whole dataset into several parts, each of which is then conducted kNN classification. We conduct sets of experiments on big data and medical imaging data. In this paper we study how to efficiently perform set-similarity joins in parallel using the popular MapReduce framework. We propose a 3-stage approach for end-to-end setsimilarity joins. We take as input a set of records and output a set of joined records based on a set-similarity condition. We efficiently partition the data across nodes in order tobalance the workload and minimize the need for replication. We study both self-join and R-S join cases, and show how to carefully control the amount of data kept in main memory on each node. We also propose solutions for the case where, even if we use the most fine-grained partitioning, the data still does not fit in the main memory of a node. We report results from extensive experiments on real datasets, synthetically increased in size, to evaluate the speedup and scaleup properties of the proposed algorithms using Hadoop.

Keyword: Classification, kNN, Big data, MapReduce, Performance Evaluation

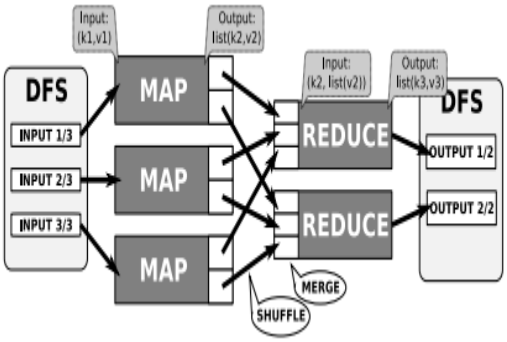
I. INTRODUCTION

In the era of big data, it is the most important to efficiently learn from large scale in all kinds of real applications, such as classification and clustering. Therefore, it is very obvious to scale traditional classification algorithms, such as decision trees, support vector machine, Naive Bayes, neutral network, and k nearest neighbors (kNN), so that these methods can be easily used in big data. Due to the simplicity, easy-understand and relatively high performance of kNN, this paper focuses on scaling the kNN classification into the application of big dataIt is frequently used as a classification or clustering method in machine learning or data mining. The primary application of a kNN join is k-nearest neighbor classification. Some data points are given for training,and some new unlabeled data is given for testing. The aim is to find the class label for the new points. For each unlabeled data, a kNN query on the training set will be performed to estimate its class membership. This process can be considered as a kNN join of the testing set with the training set. The kNN operation can also be used to identify similar images.To do that, description features (points in a dataspace of dimension 128) are first extracted from images using a feature extractor technique. Then, the kNN operation is used to discover the points that are close, which should indicates similar images.There are many applications that require detecting similar pairs of records where the records contain string or setbased data. A list of possible applications includes: detecting near duplicate web-pages in web crawling , document clustering, plagiarism detection , master data management, making recommendations to users based on their similarity to other users in query refinement , mining in social networking sites , and identifying coalitions of click fraudsters in online advertising

II. RELATED WORK

The basic solution to compute kNN adopts a nested loop approach, which calculates the distance between every object *ri* in *R* and *sj* in *S* and sorts the results to find the *k* smallest. This approach is computational intensive, making it unpractical for large or intricate datasets. Two strategies have been proposed to work out this issue. The first one consists in reducing the number of distances to compute, by avoiding scanning the whole dataset. This strategy focuses on indexing the data through efficient data structures. For example, a one-dimension index structure, the *B*+-Tree, is used in to index distances; adopts a multipage overlapping index structure R-Tree; proposes to use a balanced and dynamic M-Tree to organize the dataset; introduces a sphere-tree with a sphere-shaped minimum bound to reduce the number of areas to be searched; presents a multidimensional quad-tree in order to be able to handle large amount of data; develops a kd-tree which is a clipping partition method to separate the search space;

To apply for the traditional kNN method in big data, the previous literatures can be often categorized into two parts, i.e., fast finding the nearest samples and selecting representatives samples (or removing some samples) to reduce the calculation of kNN .



For instance, Zhang proposed a Certainly Factor (CF) measure to deal with the unsuitability of skewed class distribution in kNN methods . Li et al. proposed densitybased method for reducing the amount of training data . Zhao et al. proposed a new algorithm based on the use of labeled samples and add the screening process condition, it making the new algorithm in time complexity have significantly decreased, and no significant effect on algorithm result . These methods were mainly applied for fast search , dimension reduction and improving the efficiency of the algorithms.

III. SYSTEM ANALYSIS

System analysis is the act, process, or profession of studying an activity typically by mathematical means in order to define its goals or purposes and to discover operations and procedures for accomplishing them most efficiently.

Existing System : The research of kNN method has been becoming a hot research topic in data mining and machine learning since the algorithm was proposed in 1967. To apply for the traditional kNN method in big data, the previous literatures can be often categorized into two parts, i.e., fast finding the nearest samples and selecting representatives samples (or removing some samples) to reduce the calculation of kNN . For instance, Zhang proposed a Certainly Factor (CF) measure to deal with the unsuitability of skewed class distribution in kNN methods . Li et al. proposed a densitybased method for reducing the amount of training data . Zhao et al. proposed a new algorithm based on the use of labeled samples and add the screening process condition, it making the new algorithm in time complexity have significantly decreased, and no significant effect on algorithm result. These methods were mainly applied for fast search dimension. The primary application of a kNN join is k-nearest neighbor classification. Some data points are given for training, and some new unlabeled data is given for testing. The aim is to find the class label for the new points. For each unlabeled data, a kNN query on the training set will be performed to estimate its class membership. This process can be considered as a kNN join of the testing set with the training set. The kNN operation can also be used to identify similar images. To do that, description features (points in a dataspace of dimension 128) are first extracted from images using a feature extractor technique. Then, the kNN operation is used to discover the points that are close, which should indicates similar images.

Disadvantages : Indexes can hardly be scaled on high dimension data. The maintenance of the accuracy becomes another problem.The high cost (i.e., linear time complexity over the sample size) prohibits the traditional kNN method conducting kNN algorithm in the memory of a PC should be an interesting issue.

Proposed System: The decomposition of a distributed MapReduce kNN computation in different basic steps

• A theoretical comparison of existing techniques in Section 4, focusing on load balancing, accuracy and complexity aspects.

• An implementation of 5 published algorithms and an extensive set of experiments using both low and high dimension datasets.

• An analysis which outlines the influence of various parameters on the performance of each algorithm.

In this work, we have proposed an efficient kNN classification to conduct a k-means clustering to separate the whole dataset into several parts. We then conducted kNN classification for each part. To do this, we parted the conventional kNN method into two processes, namely training process and testing process. Moreover, we analyzed the suitable value for the parameters, such as m and k.

Advantages:

- Reducing the in-memory computational complexity.
- MapReduce is a flexible and scalable parallel and distributed programming paradigm.
- The proposed method achieved better performance than conventional kNN method clustering algorithm satisfying low complexity.

IV. ALGORITHM

We proposed a new algorithm based on the use of labeled samples and add the screening process condition, it making the new algorithm in time complexity have significantly decreased, and no significant effect on algorithm result . These methods were mainly applied for fast search dimension reduction , and improving the efficiency of the algorithms. kNN algorithm computes the distance between each training sample and test samples in the dataset and then returns k closest samples. Its time complexity is linearly and is guaranteed to find exact k nearest neighbors. However, the computational complexity of the linear search method is proportional to the size of the training dataset for each test sample, it is O(nd), where n is the size of the training dataset and d is the dimensionality . This complexity is expensive for big data. Since the kNN method is not training process, we propose to introduce a new training process for kNN, which blocks training dataset by a clustering algorithm with linear complexity. During the testing process, for each test sample, we find the k nearest cluster centers and conduct a clustering for each test sample, and then construct a new classification model base on each cluster . In particularly, the samples within a cluster has high similarity. Thus, comparing to the traditional kNN method, the proposed algorithm not only reduces the time complexity of kNN, but also does not add significantly effect on classification accuracy.

Algorithm1.The pseudo of LSC

Input: n data points x1; x2; ……; xn ARm; Cluster number k;

Output: k clusters;
1. Produce p landmark points using the k-means method;
2 .Construct a landmark matrix Z between data points and landmark samples, with the affinity calculated according to;
3 .Compute the first k eigenvectors of WWT, denoted by U¼ [u1, u2, ..., uk];
4. Compute V¼[v1, v2, ..., vk];
5 .Eacsh row of V is a data point and apply k-means to get the clusters.

V. PERFORMANCE EVALUTION

In this experiment, according to previous analysis, we set m¼10, k¼1. Then, we use classification accuracy and running time as the evaluations for the classification task . The shorter time and higher accuracy the algorithm is, the better the performance is. We report our experimental results in., we can observe that the proposed RC-kNN and LC-kNN improved by 7–9 times than the kNN (close to the number of cluster), in terms of time cost. In the evaluation of classification accuracy, the LC-kNN and RC-kNN is lower by 1–2.6% and 3.3–14% than kNN. Therefore, according to the experimental results, we may the conclusion that the LC-kNN works well in terms of classification accuracy and time.

Classification Accuracy and Time Cost of three algorithm on nine dataset.

Dataset	RC-kNN		LC-kNN		kNN	
	Accuracy	Time	Accuracy	Time	Accuracy	Time
USPS	0.9027	3.5589	**0.9355**	**3.7605**	0.9482	32.8764
MNIST	0.7221	2.9369	**0.8389**	**3.5504**	0.8635	24.1575
GISETTE	0.9311	23.3933	**0.9526**	**28.594**	0.9660	217.3327
LETTER	0.7892	3.2391	**0.9495**	**3.2994**	0.9518	19.8246
PENDIGITS	0.9452	2.3380	**0.9721**	**2.4056**	0.9780	7.2982
SATIMAGE	0.8603	1.2868	**0.8883**	**1.3027**	0.9065	3.5525
ADNC	0.7024	0.0310	**0.7667**	**0.0333**	0.7857	0.0355
psMCI	0.4792	0.0168	**0.5833**	**0.0171**	0.6042	0.0176
MCINC	0.5476	0.0468	**0.6159**	**0.0506**	0.6413	0.0575

VI. CONCLUSION

In this work, we have proposed an efficient kNN classification to conduct a k-means clustering to separate the whole dataset into several parts. We then conducted kNN classification for each part. To do this, we parted the conventional kNN method into two processes, namely training process and testing process. Moreover, we analyzed the suitable value for the parameters, such as m and k. Furthermore, we took the kNN as the baseline and conducted a groups of compared experiments among kNN, LC-kNN and RCkNN. In this paper, we have studied existing solutions to perform the kNN operation in the context of MapReduce. We have first approached this problem from a workflow point of view. We have pointed out that all solutions follow three main steps to compute kNN over MapReduce, namely preprocessing of data, partitioning and actual computation. We have listed and explained the different algorithms which could be chosen for each step, and developed their pros and cons, in terms of load balancing, accuracy of results, and overall complexity. In a second part, we have performed extensive experiments to compare the performance, disk usage and accuracy of all these algorithms in the same environment.

REFERENCE

[1] K. Inthajak, C. Duanggate, B. Uyyanonvara, S. Makhanov, and S. Barman, "Medical image blob detection with feature stability and knn classification," in *Computer Science and Software Engineering*, 2011.

[2] F. Korn, N. Sidiropoulos, C. Faloutsos, E. Siegel, and Z. Protopapas, "Fast nearest neighbor search in medical image databases," in *VLDB'96*, 1996.

[3] H. V. Jagadish, B. C. Ooi, K.-L. Tan, C. Yu, and R. Zhang, "idistance: An adaptive b+-tree bas ed indexing method for nearest neighbor search," *ACM Trans. Database Syst.*, vol. 30, no. 2, pp.364–397,2005.

[4] C. Bohm and F. Krebs, "The k-nearest neighbour join: Turbo ¨ charging the kdd process," *Knowl. Inf. Syst.*, vol. 6, no. 6, pp. 728– 749, Nov. 2004.

[5] P. Ciaccia, M. Patella, and P. Zezula, "M-tree: An efficient access method for similarity search in metric spaces," in *VLDB'97*, 1997.

[6] C. Yu, R. Zhang, Y. Huang, and H. Xiong, "High-dimensional knn joins with incremental updates,"*GeoInformatica*,2010.

[7] J. L. Bentley, "Multidimensional binary search trees used for associative searching," *Commun.ACM*,1975.

[8] J. Dean and S. Ghemawat, "Mapreduce: Simplified data processing on large clusters," *Commun.ACM*,2008.

[9] G. Song, J. Rochas, F. Huet, and F. Magoules, "Solutions for ` Processing K Nearest Neighbor Joins for Massive Data on MapReduce," in *23rd Euromicro International Conference on Parallel, Distributed and Network-based Processing*, Turku, Finland, Mar. 2015.

[10] W. Liu, J. He, S. Chang, Large graph construction for scalable semi-supervised learning, in: Proceedings of the 27th International Conference on Machine Learning, 2010.

[11] X. Wu, C. Zhang, S. Zhang, Database classification for multi-database mining, Inform. Syst. 30 (1) (2005) 71–88.

[12] X. Wu, S. Zhang, Synthesizing high-frequency rules from different data sources, IEEE Trans. Knowl. Data Eng. 15 (2) (2003) 353–367.

[13] X. Wu, C. Zhang, S. Zhang, Efficient mining of both positive and negative association rules, ACM Trans. Inform. Syst. 22 (3) (2004) 381–405.

[14] S. Zhang, KNN-CF approach: incorporating certainty factor to knn classification, IEEE Intell. Inform. Bull. 11 (1) (2010) 24–33.

[15] Y. Zhao, S. Zhang, Generalized dimension-reduction framework for recentbiased time series analysis, IEEE Trans. Knowl. Data Eng. 18 (2) (2006) 231–244.

