Nearest Neighbor Set Search With Keyword

[1]Naik-Nimbalkar S.D, [2]Dr.B.M.Patil
[1]M.Tech Student, [2]Dean of P.G.
[1]Department of P.G.,
[1]College of Engineering, Ambajogai, India.

Abstract : Mining massive and high-speed data streams among the main contemporary challenges in now days. This calls for methods displaying a high computational efficacy, with ability to continuously update their structure and handle ever-arriving big number of instances. In this work, we present a new distributed classifier based on the popular nearest neighbor concept. This method includes a technique to perform a search operation with the help of keyword. Additionally, we propose an efficient incremental instance selection method for massive data streams that continuously update and remove outdated examples from the case-base. This alleviates the high computational requirements of the original classifier, thus making it suitable for the considered problem. Experimental study conducted on a set of real-life massive data streams proves the usefulness of the proposed solution and shows that we are able to provide the first efficient nearest neighbor solution for high-speed big and streaming data.

IndexTerms **– Nearest Neighbor, Data streams, Keyword Search.**

I. INTRODUCTION

The massive volume of information gathered by contemporary systems became omnipresent, as many research activities require collecting increasingly huge amounts of data. For instance, Large Hadron Collider experiments1 generates 30 peta bytes of information per year. Potential applications for massive data analysis techniques could be found in each human activity domain. Enterprises would like to discover interesting client behavior characteristics, e.g., on the basis of sensor or Internet data. Works on personalized medical treatment for individual patients based on his/her clinical records, such as medical history, genomic, cellular, and environmental data may serve as another example. We are surrounded by enormous volumes of data arriving continuously from different sources. Therefore, one may say that we are living in the *big data era*. Big data is usually characterized by the so-called 5V's (volume, velocity, variety, veracity, and value), describing its massive volume, dynamic nature, diverse forms, different qualities, and usefulness for human beings. In many cases we do not deal with static data collections, but rather with dynamic ones. They arrive in a form of continuous batches of data, known as data streams. In such scenarios, we need not only to manage the volume but also the velocity of data, thus constantly updating and adapting our learning. To add a further difficulty, many modern data sources generate their outputs with very short intervals, thus creating the issue of high-speed data streams. Massive data must be explored efficiently and converted into valuable knowledge which could be used by enterprises (among others) to build their competitive advantage.

However, there exist a considerable gap between contemporary processing and storage capacities, which demonstrates that our ability to capture and store data has far outpaced to process and utilize it. Moore's law says that processing capacity double every 18 months, while disk storage capacity doubles every 9 months (storage law). This leads to creation of the so-called *data tombs*, i.e., volume of data which are stored but never analyzed. Therefore, we have to develop dedicated tools and techniques which are able to mine enormous volumes of incoming data, while additionally taking into consideration that each record may be analyzed only once to reduce the overall computing costs.

II. LITERATURE REVIEW

- **"Data mining with Big Data"**

Wu.zhu[19] and every in this paper have discussed concept of big data related terms and sources of data. They presented HACE theorem to characterize the features of big data. Depending on these features big data processing model deigned which deals with data mining. One more model is developed for aggregation of information from distributed sources. Paper doesn't include algorithms for processing a data rather it consider the challenges in big data and analyze the issues.

- **"Scalable Nearest Neighbor Algorithm For High Dimensional Data"**

Muja and Lowe[22] , has developed to new algorithms namely randomized k-d forest and priority search means tree to find nearest neighbor matches to high dimensional vectors in training data, as well as, In this paper distributed nearest neighbor matching framework is used for algorithms to work on large datasets paper address an issue related to scaling very large size data sets.

- **"Nearest Keyword Set Search in Multi-Dimensional Datasets"**

Singh , Zong[36] and every in paper "Nearest Keyword Set Search in multi-dimensional Datasets" considered objects which are tagged with keywords. They have proposed a method called ProMish. Which uses random projection and hash based structures. The algorithm implemented hase is go time speedup over tree based technique. ProMish has provided solution for top-k nearest keyword set search in multi-Dimensional datasets. Totally based on index in which ProMish finds optimal subset of points and ProMish- A search near optimal results.

- **"Fuzzy Based Scalable Clustring Algorithms For Handling Big Data Using Apache Spark"**

Bharill , Tiwari[37] and every in this paper concerned clustering framework by Apache spark and developed algorithm scalable Random sampling with iterative optimization Fuzzy C-means which implemented on Apache Spark cluster. This algorithm handles challenges in big data clustering. The algorithm basically works on strategy of dividing data into chunks and then process the data points within the chunks in parallel. The author focused on big data processing in terms of clustering.

- **Nearest Neighbor Classification for High-Speed Big Data Streams Using Spark**

Krawczyk , Garcia[40] and every in the paper 'Nearest Neighbor Classification for High-Speed Big Data Streams Using Spark' has given approach incremental and distributed classifier for mining massive and high speed data strems, they have provided solution using Apache Spark which includes DS-RNGE solution for processing massive streams. The DS-RNGE includes instance selection technique to improve performance by allowing insertion of correct example and removes outdates ones.

III. METHODOLOGY

The Methodology can be seen by fallowing Architectural Designed for Insertion of new arriving element in database.The first part shows how top tree is built. Ex. In first initialization there are two partitions e1 and e2 called partition -1 and partition -2 respectively. In the next step suppose a new data element arrived called as e3 then decision needs to be taken where to insert this newly arrived element. The nearest neighbor search is performed at top tree then it's been decided weather to root that element to left or to right partition of root i.e Top Tree. The Nearest neighbor algorithm gives result for insertion of a newly arrived element. At the end element is inserted to nearest neighbor partition. Whenever a new data arrives its first compared with top tree.

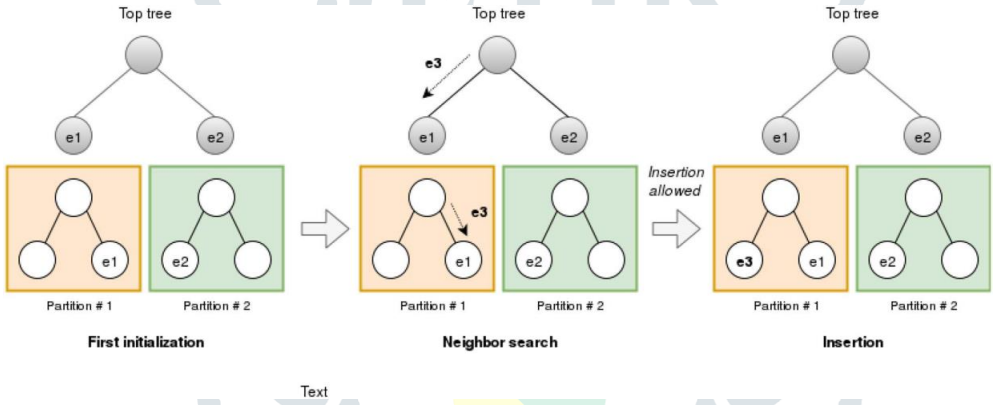
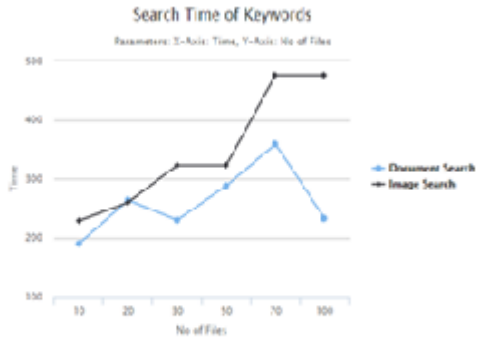


Figure: Architectural View

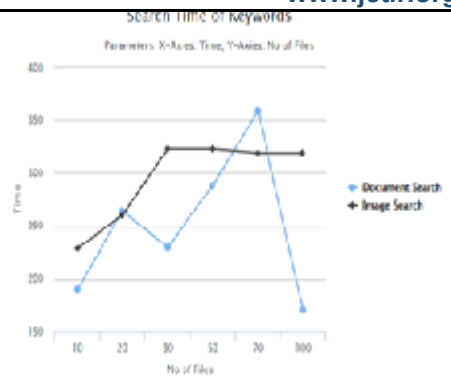
There are a lot of use cases for a system described in the introduction, but the focus of this post will be on data processing – more specifically, batch processing. Batch processing is an automated job that does some computation, usually done as a periodical job. It runs the processing code on a set of inputs, called a batch. Usually, the job will read the batch data from a database and store the result in the same or different database. Dataset D, Maximum Iterations Imax, L list of n elements with records, T Target value.

IV. RESULT

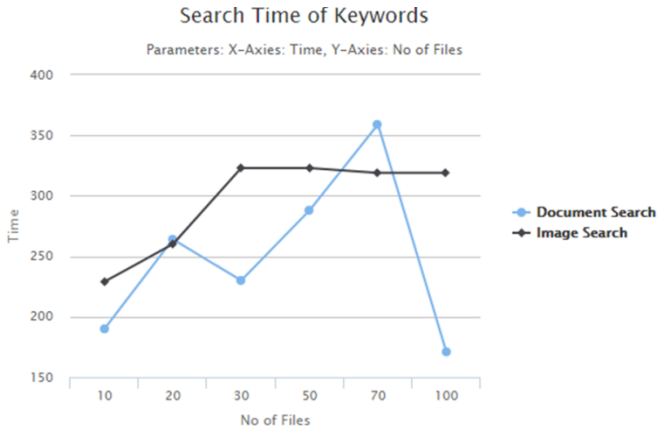
The project work can only be seen by analyzing the results. Here in Fallowing results aggregated based on number of files keywords with search time of keyword. The Lines with the blue showing graph for Document search and the line with black showing result for Image search. In the above graph, Initial result are generated as per number of files and time. Initially we have added of files to our database, the files are document as well as Image files there is an Hike in graph after insertion of 10-20 files. In the next step, we again have made insertion of few more file. Now, we can observe there is hike in graph between 20 to 30 the graph is decreased. For document search the same observations carried out for image search.



(a)



(b)



(c)

Finally we can make a conclusion that even though the files are incremented, after a peak point we are getting constant search time for document as well as Image search, in spite of increase in number of files.

ACKNOWLEDGMENT

In this paper, we propose an efficient nearest neighbor solution classify high-speed and massive data streams. Our algorithm consists of a distributed case base and an instance selection method that enhances its performance and effectiveness. A distributed metric tree has been designed to organize the case-base and consequently to speed up the neighbor searches. This distributed tree consists of a top-tree that routes the searches in the first levels and several leaf nodes that solve the searches in next levels through a completely parallel scheme. Performance is further improved by a distributed edition-based instance selection method, which only inserts correct examples and removes the noisy ones. Up to the best of our knowledge, this is the first lazy learning solution in dealing with large-scale, high-speed, and streaming problems.

CONCLUSION

we presented a new incremental and distributed classifier based on the popular nearest neighbor algorithm, adapted to such a demanding scenario. This method, includes a distributed metric-space ordering to perform faster searches. Additionally, we propose an efficient incremental instance selection method for massive data streams that continuously update and remove outdated examples from the case-base. In many cases we do not deal with static data collections, but rather with dynamic ones. They arrive in a form of continuous batches of data, known as data streams. In such scenarios, we need not only to manage the volume but also the velocity of data, thus constantly updating and adapting our learning. To add a further difficulty, many modern data sources generate their outputs with very short intervals, thus creating the issue of high-s peed data streams.

REFERENCES

[1] R. O. Duda and P. E. Hart, Pattern Classification and Scene Analysis, vol. 3. Hoboken, NJ, USA: Wiley, 1973.
[2] S. Guha, R. Rastogi, and K. Shim, “Cure: An efficient clustering algorithm for large databases,” in Proc. ACM Record, 1998, vol. 27, no. 2, pp. 73–84.
[3] A. K. Jain, M. N. Murty, and P. J. Flynn, “Data clustering: a review,” ACM computing surveys (CSUR), vol. 31,no. 3, pp. 264–323, 1999.
[4] J. Shi and J. Malik, “Normalized cuts and image segmentation,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 22, N o. 8, pp. 888–905, Aug. 2000.
[5] M. Steinbach, G. Karypis, and V. Kumar, “A comparison of document clustering Workshop Text Mining, 2000, vol. 400, no. 1, pp. 525–526.
[6] R. Krishnapuram, A. Joshi, O.Nasraoui, and L. Yi, “Low-complexity fuzzy relational clustering algorithms for Web

mining," IEEE Trans. Fuzzy Syst., vol. 9, no. 4, pp. 595–607, Aug. 2001.
[7] A. Ng, M. Jordan, and Y. Weiss, "On spectral clustering: Analysis and an algorithm," in Proc. Advances Neural Inf. Process. Syst., 2002, vol. 2, pp. 849–856.
[8] M. E. J. Newman and M. Girvan, "Finding and evaluating community structure in networks," Phys. Rev. E, vol. 69, 2004, Art. no. 026113.
[9] S. Har-Peled and S. Mazumdar, "On coresets for k-means and k-median clustering," in Proc. 36th Annu. ACM Symp. Theory Comput., 2004, pp. 291–300.
[10] T. C.Havens, J. C. Bezdek, C. Leckie, L. O.Hall, andM. Palaniswami, "Fuzzy c- means algorithms for very large data," IEEE Trans. Fuzzy Syst., vol. 20, no. 6, pp. 1130–1146,Dec. 2012.means," in Proc. IEEE Int. Conf. Fuzzy Syst., 2007, pp. 1–7
[11] M. Filippone, F. Camastra, F. Masulli, and S. Rovetta, "A survey of kernel and spectral methods for clustering," Pattern recognition, vol. 41, no. 1, pp. 176–190, 2008.
[12] J. Dean and S. Ghemawat, "MapReduce: Simplified data processing on large clusters," Commun. ACM, vol. 51, no. 1, pp. 107–113, 2008.
[13] L. Kaufman and P. J. Rousseeuw, "Finding Groups in Data: An Introduction to Cluster Analysis", vol. 344. Hoboken, NJ, USA: Wiley, 2009.
[14] P. Hore, L. O. Hall, D. B. Goldgof, Y. Gu, A. A. Maudsley, and A. Darkazanli, "A scalable framework for segmenting magnetic resonance images," J. Signal Process.Systems, vol. 54, no. 1–3, pp. 183–203, 2009.
[15] N. Labroche, "New incremental fuzzy c medoids clustering algorithms," in Proc. Annu.Meeting North Amer. Fuzzy Inf. Process. Soc., 2010, pp. 1–6.
[16] F. Nie, Z. Zeng, I. W. Tsang, D. Xu, and C. Zhang, "Spectral embedded clustering: A framework for in-sample And out-of-sample spectral clustering," IEEE Trans. Neural Netw., vol. 22, no. 11, pp. 1796–1808, Nov. 2011.
[17] F. Nie, D. Xu, and X. Li, "Initialization independent clustering with actively self- training method," IEEE Trans. Syst. Man Cybern. Part B (Cybern.), vol. 42, no. 1, pp. 17–27, Feb. 2012.
[18] J. C. Bezdek, Pattern Recognition with Fuzzy Objective Function Algorithms.
[19] Xindong Wu, Fellow, IEEE, Xingquan Zhu," Data Mining with Big Data" IEEE Trans Big Data. vol. 26, no. 1, pp.97-107,Jan. 2014.
[20] N. Bharill and A. Tiwari, "Handling big data with fuzzy based classification approach," in Advance Trends in Soft Computing. Berlin, Germany: Springer, pp. 219–227. 2014
[21] M. Han, M. Yan, J. Li, S. Ji, and Y. Li, "Neighborhood-based uncertainty generation in social networks," J. Combinatorial Optimization, vol. 28, pp. 561–576, 2014
[22] Marius Muja, David G. Lowe, Member of IEEE "Scalable Nearest Neighbor Algorithms for High Dimensional Data", IEEE Transactions on Pattern Analysis and Machine Intelligence , vol. 36, no.11, pp.2227-2240 , November 2014.
[23] R. Hong, Y. Yang, M. Wang, X. Hua, "Learning Visual Semantic Relationships for Efficient Visual Retrieval," IEEE Transactions on Big Data, vol.1, no.4, pp.152-161, 2015.
[24] X. Tian, Y. Lu, N. Stender, L. Yang, D. Tao, "Exploration of Image Search Results Quality Assessment," IEEE Transactions on Big Data, vol.1, no.3, pp.95-108, 2015.
[25] L. Yan, H. Shen, and K. Chen, "TSearch: Target-oriented lowdelay node searching in DTNs with social network properties," in Proc. IEEE Conf. Comput. Commun., 2015, pp. 3841–3855.
[26] B. Wu and H. Shen, "Analyzing and predicting news popularity on twitter," Int. J. Inf. Manage., vol. 35, pp. 702–711, 2015.
[27] Isaac Triguero, Jesus Maillo, Julian Luengo, Salvador Garcia, and Francisco Herrera, "From Big data to Smart Data with the K-Nearest Neighbours algorithm", Journal of LATEX Class Files, Vol.14, no.8, pp 1-6, August 2015.
[29] K. Huang, C. Wang, and D. Tao, "High-order topology modeling of visual words for image classification," IEEE Trans. Image Process., vol. 24, no. 11, pp. 3598–3608, Nov. 2015.
[30] X. Tian, Y. Lu, N. Stender, L. Yang, and D. Tao, "Exploration of image search results quality assessment," IEEE Trans. Big Data, vol. 1, no. 3, pp. 95–108, Sep. 2015
[31] F. Wu, Z. Wang, Z. Zhang, Y. Yang, and J. Luo, "Weakly semisupervised deep learning for multi-label image annotation," IEEE Trans. Big Data, vol. 1, no. 3, pp. 109–122, Jul.-Sep. 2015.
[32] Y. Yang, F. Shen, H. T. Shen, H. Li, and X. Li, "Robust discrete spectral hashing for large-scale image semantic indexing," IEEE Trans. Big Data, vol. 1, no. 4, pp. 162–171, Oct.-Dec. 2015.
[33] Y. C. Wang, C. C. Han, C. T. Hsieh, Y.-N. Chen, and K.-C. Fan, "Biased discriminate analysis with feature line embedding for relevance feedback-based image retrieval," IEEE Trans. Multimedia, vol. 17, no. 12, pp. 2245–2258, Dec.2015
[34] Y. Lin and H. Shen, "VShare: A wireless social network aided vehicle sharing system using hierarchical cloud architecture," in Proc. IEEE 1st Int. Conf. Internet-of-Things Des. Implementation, 2016, pp. 37–48.
[35] M. Han, M. Yan, Z. Cai, Y. Li, X. Cai, and J. Yu, "Influence maximization by probing partial communities in dynamic online social networks," Trans. Emerging Telecommun.Technol., vol. 10, pp. 561–576, 2016.
[36] Vishwakarma Singh, Bo Zong , and Ambuj K. Singh, "Nearest Keyword Set Search in Multi-Dimensional Datasets", IEEE Transactions on Knowledge and Data Engineering, vol.28, no.3, pp. 741-755, March2016.
[37] Neha Bharill, Aruna Tiwari, Aayushi Malviya, member IEEE,"Fuzzy Based Scalable Clustering Algorithms for Handling Big Data Using Apache Spark",IEEE Transactions on big data, vol.2, no.4,pp.339-352, oct-dec 2016.
[38] Bo Wu and HaiyingShen, Member, IEEE "Exploiting Efficient Densest Subgraph Discovering Methods"IEEE Trans Big Data,vol.3,pp.334-348,Sept.2017.
[39] Ming Shao, Member, IEEE, Xindong Wu, Fellow, IEEE, and Yun Fu, Senior Member, IEEE "Scalable Neares Neighbor Sparse Graph Approximation by Exploiting Graph Structure" IEEE Trans Big Data.vol.2,pp.97- 107 Dec.2018.
[40] Sergio Ramirez-Gallego, Bartosz Krawczyk, Salvador Garcia, member ,IEEE, "Nearest Neighbor Classification for High-Speed Big Data Streams Using Spark", IEEE Transactions on system. Man. and Cybernetics System. vol.47. no.10, pp.2727-2738, October 2017
[41] Yufie Tao, Cheng Sheng, "Fast Nearest Neighbor Search with Keywords" , IEEE Transactions on knowledge and Data Engineering.pp-1-13