Journal of Applied Sciences

ISSN 1812-5654

science alert

ANSInet
an open access publisher
http://ansinet.com

Journal of Applied Sciences 13 (12): 2208-2212, 2013
ISSN 1812-5654 / DOI: 10.3923/jas.2013.2208.2212

A Fast Classification Algorithm for Big Data Based on KNN

[1]Kun Niu, [1]Fang Zhao and [2]Shubo Zhang
[1]School of Software Engineering, Beijing University of Posts and Telecommunications, 146,
10 Xi Tu Cheng Rd, 100876, Beijing, People's Republic of China
[2]Marketing Research Center, China Telecom Beijing Research Institute, Beijing, People's Republic of China

Abstract: As massive data acquisition and storage becomes increasing affordable, a wide variety of researchers are employing methods to engage in sophisticated data mining. This study focuses on fast classification for big data based on a traditional classification method KNN (K-Nearest Neighbor). We reform the standard KNN algorithm and present a new algorithm named NFC (Neighbor Filter Classification). The NFC algorithm firstly computes the class distribution in each attribute of original dataset and sorts attributes by classification contribution. Secondly, NFC gets the model of the KNN result on training set to estimate the finite scope of the k-nearest neighbor. Then NFC uses test set to get the proper parameters and updates model regularly to make it efficient. Experimental results show the excellent ability of classification and low computation cost of NFC.

Key words: K-nearest neighbor, big data, classification, data mining

INTRODUCTION

In many areas such as science, simulation, Internet and e-commerce, the volume of data to be analyzed grows rapidly. Parallel techniques which could be expanded cost-effectively should be invented to deal with the big data (Qin *et al*., 2012). Compared with traditional data warehouse applications, big data analytics are huge and complex (Wang *et al*., 2011). Relational data management technique has gone through a history of nearly 40 years. Now it encounters the tough obstacle of scalability, which relational techniques cannot handle large data easily (Qin *et al*., 2012). In the meantime, none relational techniques, such as Map Reduce as a typical representation, emerge as a new force and expand their application from web search to territories that used to be occupied by relational database systems. They confront relational technique with high availability, high scalability and massive parallel processing capability (Qin *et al*., 2012). However, Map Reduce only lifts the ability of data operating systems, not for analysis methods. The existing efficient algorithms do not work or work well on real big data sets.

Meanwhile, after being presented on IJCAI in 1989, KDD (Knowledge Discovery in Database), well known as data mining, has been popular for over twenty years. Many researchers present lots of brilliant algorithms and build the infrastructure of new field. Although data mining brings very useful information and knowledge of regular patterns, it also takes much more CPU time for complicated computing. For example, the computational complexity of the popular clustering method k-means is 0 (nkt), where number of objects is n, number of clusters is k and number of iterations is t. The computational complexity of the popular classification method ID3 and C4.5 is 0 (nm^2), where number of objects in training set is n and number of attributes is m. The computational complexity of the popular association rules mining method apriori is:

$$0\left(kt+\sum_{k\geq 2}|L_k|^2+t|C_k|+|C_k|\right)$$

in the worst case, where number of transactions is t, the longest transaction length is k, candidates set is C_k and frequent item set is L_k. These scales of computations are curse of big data, especially for stream data applications such as social network. Certainly we cannot get knowledge from the original stream data directly because it expands rapidly. However, data mining models always work for a long time and rarely update, which only fit current data set. For big data, useful patterns and knowledge changes frequently according to the fast increasing data volume. The models often lose effect in an hour even a minute. In this study, we focus in classification on big data. We reform the standard KNN (K-Nearest Neighbor) algorithm to realize fast classification operation in big data.

Corresponding Author: Kun Niu, School of Software Engineering, Beijing University of Posts and Telecommunications, 146, 10 Xi Tu Cheng Rd, 100876, Beijing, People's Republic of China

Related works: To design a favorable architecture for big data analytics (Wang *et al*., 2011) lists some key features for big data analytics, summarizes current main implementation platforms (parallel databases, Map Reduce and hybrid architectures based on them) and points their pros and cons. Shen *et al*. (2013) surveys the researches about typical NoSQL database based on key-value data model. It introduced the characteristics of big data and the key technique issues supporting big data management. It also gave the frontier efforts and research challenges including system architecture, data model, access mode, index, transaction, system elasticity, load balance, replica strategy, data consistency, flash cache, Map Reduce based data process and new generation data management system etc. Cohen *et al*. (2009) highlights the emerging practice of Magnetic, Agile, Deep (MAD) data analysis as a radical departure from traditional Enterprise Data Warehouses and Business Intelligence. However, these methods are limited to propose the whole framework and they does not work for methods or algorithms of data mining applications.

Our contributions: This study focuses on fast classification for big data. It reforms and improves the standard KNN algorithm. It presents a new algorithm named NFC. NFC firstly computes the class distribution in each attribute of original dataset and sorts attributes by classification contribution. It gets the model of the KNN result on training set to estimate the finite scope of the k-nearest neighbor. Then it uses test set to get the proper parameters and updates model regularly to make it efficient. Experimental results show the excellent ability of classification and low computation cost of NFC.

Overall, the contributions of present study for big data classification are: We propose the general policy of data mining in big data environment. We improve traditional KNN method and give the new fast classification method for big data.

NEW ALGORITHM-NFC

General policy of data mining on big data: Big data is famous because of its volume, velocity, variety and veracity. Of the four points, volume and variety are challenges of system processing capacity. It will be dealt well by tools based on Map Reduce. Because of variety, big data set has to be filtered, transformed and recoded after being extracted and before being loaded. This process is also powerfully supported by Map Reduce to cut down the cost of database. Meanwhile, velocity and veracity aim at modeling methods. Veracity requires the algorithm be accuracy, which means lower misclassification ratio of classification, a smaller average distance to cluster center and bigger average distance between cluster centers. Velocity requires the computation of modeling be rather simple. The computation process has to be finished before the next period data comes so we can make correct decisions and response to business properly.

Therefore, the data mining model on big data has to satisfy the three rules as following: Computational complexity decreases to 0(n) and in any period n is less than the data volume of coming next in modeling time. The model is able to replace or adjust key input parameters automatically when being operated. The model updates other (not key) parameters regularly.

Improvement of KNN: KNN is a popular classification algorithm. Based on the theory of linear interpolation, it judges the class of an object by its nearest k neighbors' class labels. KNN meets overwhelming bottleneck on big data as follows:

- The time cost of modeling is unacceptable. When a new object comes, it has to compute all the distances between the existing objects and the new comer. If current sample size is large enough, we cannot finish work before the next period data comes
- The model of KNN is sensitive to parameter k. It will be over fitting when k is small and too smooth when k is large
- To overcome these shortcomings, we firstly involve dimensionality reduction to filter out attributes which do not contribute much to classification. Secondly, we store the best probable neighborhood of each attribute to find one's KNN much easily. Thirdly we present an effective method of finding proper k

Algorithm description: Big data sets always increase by time. Then NFC makes a four-step work flow to build and update the model. It is shown in Fig. 1.

Dimensionality reduction. It is a well-known way to reduce the computational cost of KNN. However, it always focuses on distribution of object instead of class, which does not measure the difference effectively.

In NFC, we measure the contribution to classification by discrimination based on entropy theory.

Definition 1: EID (entropy of object distribution): Let D be a dataset with N objects. Each attribute in D is divided into k equal cells, number of objects of the i_{th} cell is n_i, then EID of an attribute is defined as following:

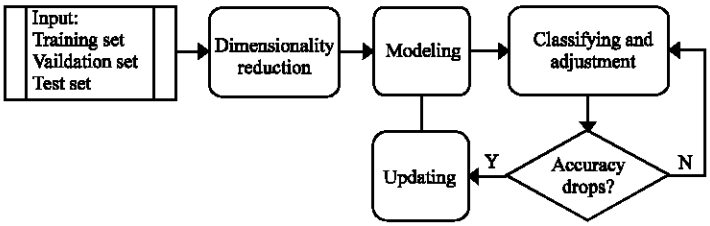

Fig. 1: Process of NFC

$$EID = -\sum_{i=1}^{k} \frac{n_i}{N} \log \frac{n_i}{N} \quad (1)$$

The EID presents the imbalance of object distribution. But it is not obviously relative to classification. So we give the ECD to select useful attributes.

Definition 2: ECD (entropy of class distribution): Let D be a dataset with N objects in p classes. Each attribute is divided to k equal cells, number of objects of the i_{tk} cell is n_i and n_{ij} is the number of object of the j_{tk} class in i_{tk} cell, then ECD of an attribute is defined as following:

$$EID = \frac{\sum_{j=1}^{p} \sum_{i=1}^{k} \frac{n_{ij}}{n_i} \log \frac{n_{ij}}{n_i}}{p} \quad (2)$$

The ECD evaluates the imbalance of class distribution. The attributes with higher ECD contribute more in classification than the lower ones because they imply more information of how to classify objects. However, in real life data set, ECD is difficult to compute because higher ECD brings errors of divided by zero. So we define the following discrimination to avoid that.

Definition 3: Discrimination: Let D be a dataset with N objects in P classes. For each attribute, objects are sorted increasingly and then divided into k equal cells by number of instances (instead of polar distance in ECD), suppose n_{ij} is the number of instances of j_{tk} class in i_{tk} cell, the discrimination of an attribute is defined as following:

$$Discrimination = -\frac{\sum_{j=1}^{k} \sum_{i=1}^{p} \frac{k \times n_i}{N} \log \frac{k \times n_{ij}}{N}}{k} \quad (3)$$

Discrimination is transformation of ECD. It allocates each cell with same volume by sorting objects to avoid computational difficulty. Fig. 2 shows the meaning of discrimination, obviously the information that class B tends to be low in current attribute is contained in discrimination.

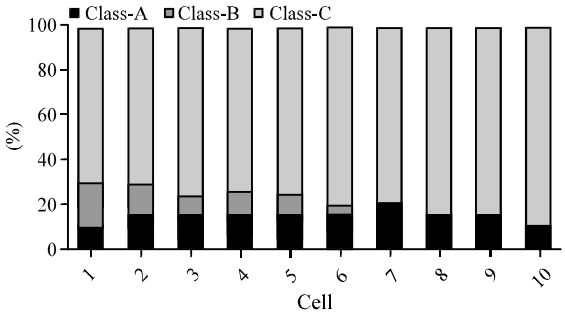

Fig. 2: Meaning of discrimination

In NFC dimensionality reduction process has four sub steps. Sorting objects increasingly in each attribute individually. Group objects to cells of equal members. Getting discrimination of each dimension. Sorting attributes by discriminations. Select the attributes with higher discriminations and delete the lower ones where the threshold is average of all.

MODELING

Before modeling, we cut the whole data set into two parts, training set and test set. Generally training set covers two-thirds objects and test set the other one-thirds. Thus we can catch the most probable neighborhood where one's KNN hide in and get a proper parameter k. The modeling way of NFC is a standard KNN but pays extra attention to locking the expecting location of each object's KNN.

Training: The objects of train set are placed into a multiple dimensional space of D like initialization of KNN. Of course training set has class labels. Then objects of test set are inputted to D one by one without class label. They find their KNN in D, be classified by the k neighbors and record the range of KNN with itself as reference. For example, (1, 1, 1) has three nearest neighbors (1, 2, 1), (0, 1, 1) and (1, 1, 2), then the ranges to be stored of the three dimensions are [-1, 0], [0, 1] and [0, 1].

However, different from KNN, in NFC each input of test set does not involved in D when the next input comes. They seem to be inputted exactly at same time.

Set parameters: After this training process, we average all the stored ranges of KNN from every object of test set. Then we get the most probable neighborhood of one's KNN which makes prospective searching much easier.

Meanwhile, we compare the original class labels of test set with result of KNN model, the best k is selected by the lowest error rate.

CLASSIFYING AND ADJUSTMENT

After modeling, we can apply the model on new data set. As a new object comes, NFC does not compute all its distances to the whole data set but filter out the data set quickly by the most probable neighborhood.

Classifying: When a new object be inputted, NFC searches the most probable neighborhood and filters out the uncorrelated objects. It is important that the attributes in filter process is ordered by their discriminations to save the time of finding KNN. For example, (1, 1, 1) is inputted into a space of [0, 10], [0, 10]×[0, 10] and the neighborhood is [-1, 0], [0, 1] and [0, 1]. Then we search scope [0, 1] in the first attribute which filter out 90% objects first, then search [1, 2] in the second and [1, 2] in the third if KNN is not locked. Even if this filtering process cannot lock KNN, it has filtered out 99.9% objects and avoid redundant computation.

Finally we find KNN of a new distance through filtering procedure and accomplish classification. The process of classifying is shown in Fig. 3.

Adjustment: As we indicate at the beginning, the mining algorithm on big data has to adjust itself to prevent causing errors when regular patterns change. In NFC, each new object corrects the neighborhoods of the attributes used in its classification. The index smoothing method is appropriate for this condition when trend varies very soon. The new neighborhood of an attribute is sum of half the old and half the new. That means p = 0.5 in index smoothing method.

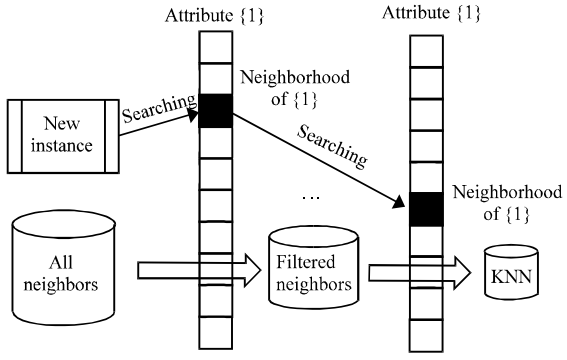

Fig. 3: Example of neighbor filtering

UPDATING

Although, the model is able to adjust itself, the parameter k stands still. The modeling procedure has to be executed again in some time. The time interval between two models depends on the probability of changing of best k. Together with the new modeling process and the new k, neighborhood of each attribute is renewed.

Process of NFC: The process of NFC is shown in Table 1.

EXPERIMENTS

To evaluate the efficiency of NFC, we have implemented NFC in Java and conducted the experiments with public datasets. The public datasets are Localization Data for Person Activity (LDPA) and Poker-hand from UCI (http://kdd.ics.uci.edu/). LDPA has 164860 objects in 11 classes and 3 attributes. Poker-hand has 1025010 objects in 10 classes and 10 attributes. Both of the two data sets are really big enough for KNN.

Experimental results show the efficiency of NFC in classification on big data set. All experiments were run on a PC with a core i5@2.5 GHz CPU and 4 GB DDR3@1600 MHz RAM.

Accuracy of classification: As shown in Table 2, NFC is better than standard KNN when k is 1, 2 and 5. For KNN,the best k is 4 in both. But in complete procedure of NFC, we find the best k = 5 through comparing labels of test set, so k cannot be 3 or 4 on the first data set. For the second data set, best k = 4 for NFC.

Table 1: Pseudo-code of NFC

Input: Data set D with M attributes and N objects
//step 1 Dimensionality Reduction
For (i=1 to M) do { Computing discrimination of each attribute {i}};
If (Discrimination(i)≤average) then {reject {i} from D};
//After fileting process, M is reduced to m.
//step 2 Modeling
For (i=1 to 2*N/3) do //training set and test set
For (j=1 to N/3) do{newlabel(j)=label(KNN(i))};
K=compare(newlabel(j),label(j));
For (i=1 to m) Neighborhood(i)=average(neighborhood(KNN)$_i$);
//Step3 Classifying and Adjustment
For each new object A do
For (i=1 to m) do //m attributes can be used
{ Filter N by neighborhood(i); If(filtered(N)=k){label(k)=label(KNN); neighborhood(i)=0.5*neighborhood(i)+0.5*(KNN$_i$-A$_i$); }
Else i=i+1}
//step 4 Undating
If (error rate=average*1.05%) {Go step 2;} Else {go step 3;}

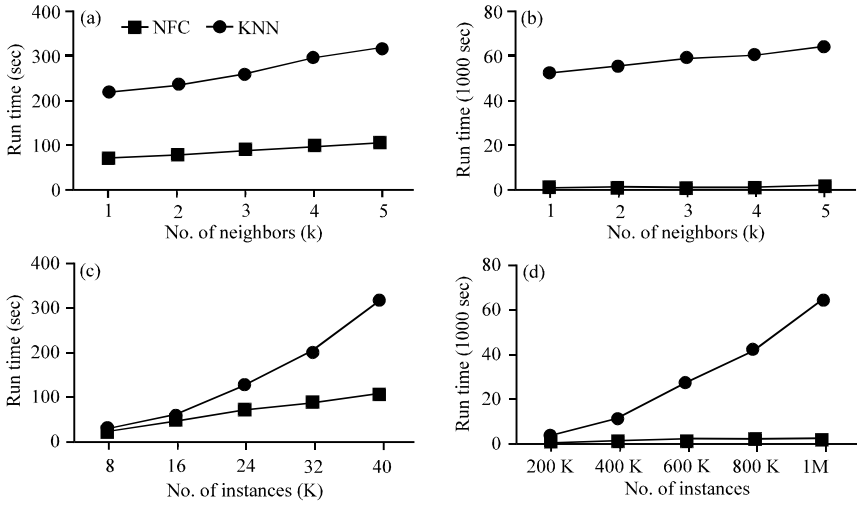

Fig. 4: CPU time of NFC and KNN

Table 2: Accuracy of classification

		Precision of algorithms		
Data set	k	NFC (%)	KNN (%)	Disparity (%)
Localization data for person activity	1	41.03	40.92	+0.11
	2	41.36	40.92	+0.44
	3	43.62	44.83	-1.21
	4	44.86	45.93	-1.07
	5	45.57	42.65	+2.92
Poker-hand	1	51.11	51.06	+0.05
	2	53.63	51.06	+2.57
	3	53.65	53.13	+0.52
	4	55.03	54.61	+0.42
	5	54.95	54.58	+0.37

Computational complexity: To evaluate the scalability of the new algorithm, we test the run time of NFC and KNN on two data sets. As shown in Fig. 4(a, b), the run time of both NFC and KNN increased almost linearly with the increasing of k. However, Fig. 4(c, d) show that NFC displays a linear complexity with much better scalability than KNN in both two data sets against the number of instances.

CONCLUSION

In this study, we address the problem of fast classification for big data and provide a new algorithm named NFC. It firstly computes the class distribution in each attribute of original dataset and sorts attributes by classification contribution. Secondly, NFC gets the model of the KNN result on training set to estimate the finite scope of the k-nearest neighbor. Then NFC uses test set to get the proper parameters and updates model regularly to make it efficient. Experimental results show the excellent ability of classification and low computation cost of the new method. As future work, we will extend our method to also handle more applications in big data mining.

ACKNOWLEDGMENT

This study was supported by the National Key Technology R and D Program of the Ministry of Science and Technology of China (2012BAJ18B07-05, 2012BAJ18B08), the Nation Natural Science Foundation of China (61374214) in part, the Major Projects of Ministry of Industry and Information Technology (2010ZX03006-00203, 2011ZX03005-005) in part, the Electronic Information Industry Development Fund Project of Information Industry Department (2012-380), Tianjin Binhai New Area Science Little Giant Enterprises Growth Plan (2011-XJR12009), National Key Basic Research Program of China (973 Program)(2012CB315802), National Natural Science Foundation of China(61171102) and the Fundamental Research Funds for the Central Universities(2013RC0501).

REFERENCES

Cohen, J., B. Dolan, M. Dunlap, J.M. Hellerstein and C. Welton, 2009. MAD skills: New analysis practices for big data. VLDB, Lyon, France, August 24-28, 2009. VLDB Endowment Inc., pp: 1481-1492.

Qin, X.P., H.J. Wang, X.Y. Du and S. Wang, 2012. Big data analysis-competition and symbiosis of RDBMS and MapReduce. J. Software, 23: 32-45.

Shen, D.R., G. Yu, X.T. Wang, T.Z. Nie and Y. Kou, 2013. Survey on NoSQL for management of big data. Chin. J. Soft., 24: 1786-1803.

Wang, S., H.J. Wang, X.P. Qin and X. Zhou, 2011. Architecting big data: Challenges, studies and forecasts. Chin. J. Comp., 34: 1741-1752.