Big data searching optimization with machine learning and parallel computing

Wang, Lei

(出版者 / Publisher)
法政大学大学院情報科学研究科
(雑誌名 / Journal or Publication Title)
法政大学大学院紀要. 情報科学研究科編 / 法政大学大学院紀要. 情報科学研究科編
(巻 / Volume)
9
(開始ページ / Start Page)
45
(終了ページ / End Page)
50
(発行年 / Year)
2014-03
(URL)
https://doi.org/10.15002/00010520

Big data searching optimization with machine learning and parallel computing

Wang Lei
Graduate School of Computer and Information Sciences
Hosei University
Tokyo, Japan
Email: lei.wang.2v@stu.hosei.ac.jp

Abstract*—In recent years, Internet is in the period of information explosion and data is becoming huge and complex. How to search a result efficiently from the data group, which called big data, is a problem many fields faced on. This paper describes combining machine learning, Data Mining and search index optimization based on distributed system to improve the searching efficiency and accuracy for big data. However, the machine learning processing cannot find the existed destination directly according to the query information. The classification of supervised machine learning can do a prediction after learning from training dataset, which extracted by data mining processing and data mining also helps to analysis the statistical information about the original dataset to define priority of matching steps and indexing structure. According to the prediction, searching procedure just focus on the specific classification preferentially. In this way, it is not necessary to search all data index in one query processing. So the main point is aim to reduce unconcerned information as much as possible and do a result assuming correctly. At last, the experiment on a common big data dataset, which often utilized for machine learning research, proved that the efficiency and accuracy improved by processing with 6 processors with parallel computing design and search indexing optimization. In that kind of approach to search big data, accuracy of machine learning algorithm has a direct and significant influence with dataset. So to apply this approach, the preview analysis is essential to be done.

Keywords: big data; machine learning; distributed system; parallel computing; data mining; Jubatus; zookeeper

I. INTRODUCTION

In recent years, machine-learning technology is developing rapidly and applying in many fields. It becomes more and more important in computer science, especially in big data analysis and prediction. At the same time, distributed system architecture is becoming the main approach to deal with big data instead of super-computer, running on commodity hardware. So when trying to search some data from the complex big data resources, it is not sensible to search all the big data on every query request, especially in real-time response requirement system. As we known, Google, Facebook, Amazon etc. have lots of uses, so a huge data is ready to be managed. Not only the big companies and enterprises but also scientific institutes harvest an amazing amount of data. Traditional database management systems are often unable to cope with this. The term "big data" is defined as a huge amount of digital information, which is too big and complex for normal database technology to process it, such as SQL server, Oracle Server, MySQL. In order to deal with big data many new technologies have been developed, like NoSQL (not only SQL), Hadoop, MapReduce, FileSystem. These systems are distributed system cooperation with a lot of computers instead of super-computer to reduce cost. The distributed system and parallel computing are the popular and mature techniques to deal big data.

Machine learning theory, a branch of artificial intelligence, is about the construction and study of systems that can learn from data to recognize pattern. The machine learning is now popular in prediction, it learns from data and recognizes the pattern of that kind of data. In the other hand, the overlap technology with machine learning is data mining. These two terms are commonly confused, as they often employ the same methods and overlap significantly. But they can be roughly defined as: Machine learning focuses on prediction, based on known properties learned from the training data. Data mining focuses on the discovery of (previously) unknown properties on the data. The machine learning needs to learn from training data and better training data would tremendously increase the efficiency of learning while the bad training data dose a significant influence. The data mining technology is aimed to discovery the useful data and its effect. But the machine learning accuracy is limited by lots of factors. In order to verify the result of machine learning algorithms, the parallel computing and probability theory could do a great favor to verify the results by comparing the confidence degree. I attempt to use those technologies and its characteristics for big data processing, because the machine learning can do a classifier and prediction according to input information from training data experiment. Some searching engine trend to import the machine learning for pattern detection. The searching needs more intelligent way to go.

Aiming to combine machine learning and parallel computing together, the Jubatus seems to be the most adaptive tool to implement. Various machine learning algorithms supporting, scale-up ability, real-time processing in main memory, etc. those intelligent features make Jubatus to be a powerful tool. Approach will be experiment on two datasets of NLP (Natural language processing): Adult dataset [1] and Multi-Domain Sentiment Dataset (version 2.0) [2], which

[1] http://archive.ics.uci.edu/ml/datasets/Adult
[2] http://www.cs.jhu.edu/~mdredze/datasets/sentiment/

contains product reviews taken from Amazon.com from many product types (domains). Using the regular expression to filter the original data and extract the useful data. Do a statistic information record for next analysis. To increase performance of locating destination information, the searching indexing is an essential part for searching technology. To build the optimized indexing, the data mining would do a better work on it. Machine learning to predict the target result of searching, to improve the confidence degree. According to the hypothetic result as an evidence to search indexing and locate final result.

II. Related Technologies And New Solutions

A. *Jubatus framework [1]*

Jubatus is a distributed processing framework and streaming machine learning library. It is an open source project published by NTT Corporation. Jubatus aims to achieve scalable distributed computing for profound analytical processing such as online machine learning algorithms and provide a common framework for different supported algorithms. It is the main implementation of the searching system for processing. But Jubatus is just focusing on big data processing with machine learning and distributed cooperation. I attempted to make use of the high processing efficiency and parallel computing feature for big data searching optimization based on the processing ability of Jubatus framework.

B. *Data mining*

The manual extraction of patterns from data has occurred for centuries. Data mining is the process of applying these methods with the intention of uncovering hidden patterns in large data sets. It is used to analyze the big dataset. The complex data structure should be re-organized before analyzing; knowing the features and patterns of the data could provide the information for efficient processing.

C. *Search engine indexing*

Search engine indexing collects, parses, and stores data to facilitate fast and accurate information retrieval. A suitable index facilitates efficient performance of result retrieval and less dispensable resource cost. The purpose of storing an index is to optimize speed and performance in finding relevant documents for a search query. The more searching indexing information provides, the more fast the matching the queries would have. Use the space to obtain less time cost.

D. *Apache Zookeeper[2]*

Apache Zookeeper is an effort to develop and maintain an open-source server, which enables highly reliable distributed coordination. It is a centralized service for maintaining configuration information, naming, providing distributed synchronization, and providing group services. Jubatus can run in a distributed environment using ZooKeeper and Jubatus keepers. In that kind of distributed system cluster, every machine collaborates with one another. Even multiple server processes on the same machine, the Jubatus keepers can also handle that kind of situation. Zookeeper plays the role of task management of the system. In order to handle and process the final results of processors, the Zookeeper manages results collection and output, which the Jubatus keeper does not.

III. Procedures

When the input data for search queries received by processors, the matching with index is not begin directly as the normal searching. The first step is doing a prediction by several processors according to the input data. Then eliminate the suspect prediction results by comparing to all predictions and pick up the most dependable one for the final prediction result. After the prediction is confirmed, the matching with index begins. If the matching were positive, it would locate the result and return it immediately as the searching result. Otherwise, the return is null and mean there is no record satisfies. The searching optimization procedure is shown in the Figure.1 below.

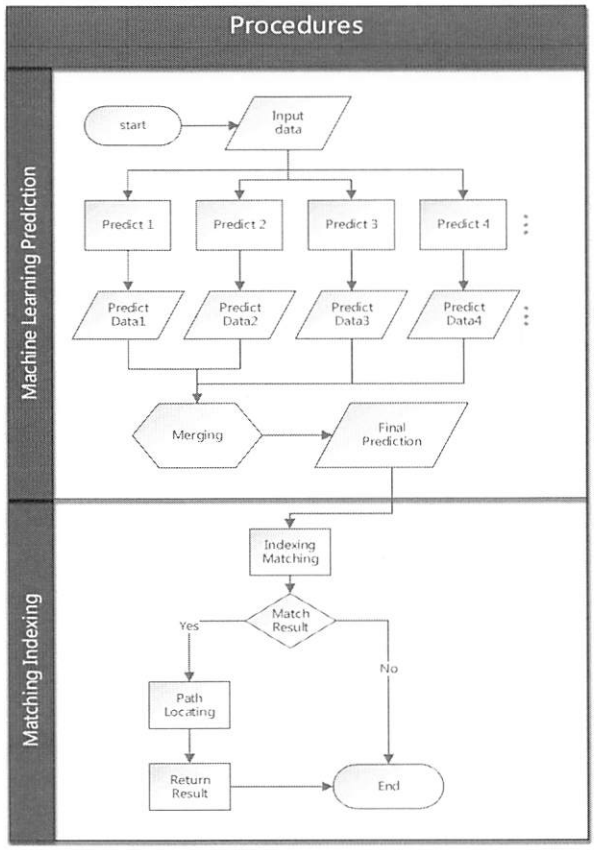

Fig. 1. The searching procedure

IV. Approach And Designing

A. *Property extraction and data statistic analysis*

In order to obtain the information about dataset, dataset analysis is a necessary preview work. Because big data is complex and informal for processing directly, a pre-process is essential for next processing. Removing the impurity information and data, which called extraction, should be done before searching index building.

B. *Searching index building*

After the properties extraction and information statistics, searching index for the particular dataset is ready to build. Attributes have two kinds of type, the Positive-Negative type and the multiple value type.

1) The positive-negative value type attribute

When the attributes are positive-negative type, applying Entropy (information theory) [3] to measure that attribute's priority for determining searching degree. The Entropy formula is defined like below:

$$Entropy(X) = -\sum_{i} p(x_i)\, log_b(x_i) \tag{1}$$

2) The multiple values type attribute

While the attributes are multiple values type, applying probability of occupation on set to measure each attribute's priority for its feature/attribute searching degree.

Sorting the features/attributes by probability value from large to small. In order to divide the set to symmetrical subsets, merging some attributes to a subset as integration should be handled.

$$\bar{P}(x) = \frac{100\%}{attributes(x)} \tag{2}$$

The $\bar{P}(x)$ is the average probability of set X, attributes (x) is the number of the set X. there is an estimation for merging attributes.

First select the attributes which need to be merged if the attribute's probability $P(x_i) < \bar{P}(x)$. So there is a subset contains $Z = \{P(x_i) | P(x_i) < \bar{P}(x)\}$.

Second, The next processing is to merge the elements in Z to a multi-attributes probability. The merging begins from the least one in the Z until the merged $P_M(x_i) \geq \bar{P}(x)$. Repeat the steps until each element in subset Z not less than $\bar{P}(x)$.

The indexing building would like a fractal tree merging [4] shown as Figure.2 below.

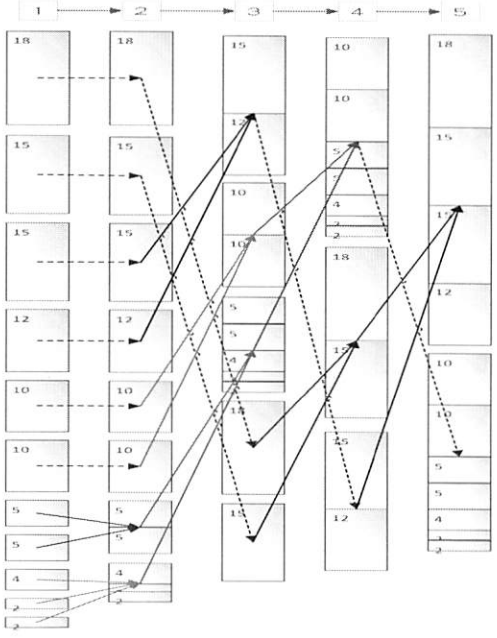

Fig. 2. Indexing building preview(UNIT %)

C. Prediction of machine learning

As we know, machine learning technique can be trained on a kind of objects to learn to distinguish new object whether the new object is belong to the training data's kind. This kind of machine learning requires training data for generalization, than evaluate new object with the generalization standard to classify the new object. There are kinds of machine learning algorithms for different requirements, such as classification, regression, recommendation and anomaly detection, etc.

1) Supported algorithms of Jubatus framework

The Jubatus supports kinds of classification algorithms in machine learning, and processing the same data with the same train data, the accuracies presents differently. There are five machine learning classification algorithms supported by Jubatus now:

- *Perceptron [5]*
- *Online passive-aggressive algorithms (PA) [6]*
- *Confidence Weighted Learning (CW)[7]*
- *Adaptive Regularization of Weight Vector [8]*
- *Normal HERD (NHERD) [9]*

2) Algorithms comparing

To compare those algorithms, the new algorithm performances better accuracy and high speed convergence than the old one normally. But in different issues, each algorithm has their own advantages. So to a particular issue, to test results are the best demonstrations.

a) Accuracy comparing

The accuracy and average rate and average cost time of correct result on Adult Data Set is represented below in Table.1 and histogram of accuracy Fig.4 according to Table.1.

TABLE.1 ACCURACIES OF MACHINE LEARNING ALGORITHMS.

Name	Accuracy_1 (%)	Accuracy_2 (%)	Accuracy_3 (%)	Average (%)	AVG Time (s)
PA	75.62	74.94	75.08	75.21	4.40
PA1	75.63	74.94	75.32	75.30	4.41
PA2	75.63	75.08	74.94	75.22	4.41
Perceptron	23.03	17.34	11.56	17.31	4.38
NHERD	74.27	74.80	75.09	74.72	4.34
CW	72.21	66.98	65.63	68.27	4.40
AROW	79.80	80.35	80.29	80.15	4.42

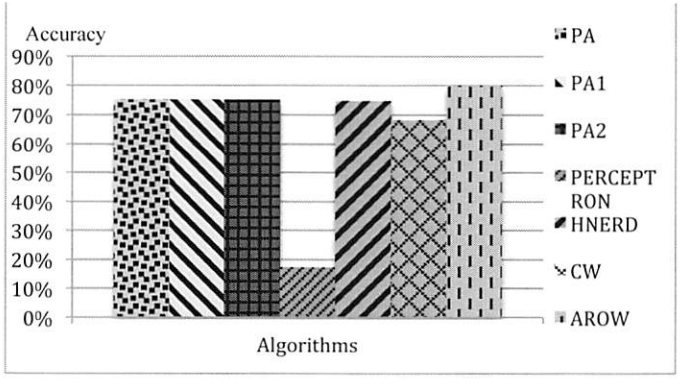

Fig. 3. Histogram of accuracy with different algorithms

Stability: We could find in the Table.1 that comparing with each other, those algorithms appear well stable nature and achieve fluctuations of errors percent is under 1% of processing except the perceptron algorithm on this kind data classification.

Efficiency: comparing the average time costs, the efficiency is very approximate and they also have a high efficiency in processing data. The high efficiency is due to the main memory processing technology of the Jubatus supports.

Accuracy: in this side, we can recognize that the AROW obtains a better performance than others, while the stability and efficiency own the approximate level to others. In the Machine learning area the 80% percent accuracy is well performance on test processing.

b) Relationship with training data attributes

- *7-Attritubes*

Test (%)	1	2	3	4
Results	79.80	80.07	80.29	80.35
5	6	7	8	Avg
80.38	80.40	80.44	80.44	80.27

- *9-Attritubes*

Test (%)	1	2	3	4
Results	82.81	83.01	83.10	83.16
5	6	7	8	Avg
83.21	83.24	83.32	83.34	83.15

- *11-Attributes*

Test (%)	1	2	3	4
Results	83.38	83.42	83.43	83.44
5	6	7	8	Avg
83.42	83.24	83.42	83.42	83.42

With the full attributes in datum format as below:

- *14-Attritubes*

Test (%)	1	2	3	4
Results	80.43	80.78	82.23	82.45
5	6	7	8	Avg
82.48	82.51	82.51	82.53	81.99

Using the experiment result with different number attributes processing to draw line chart for representing the affect of attribute number. The line chart Fig.4 is shown below:

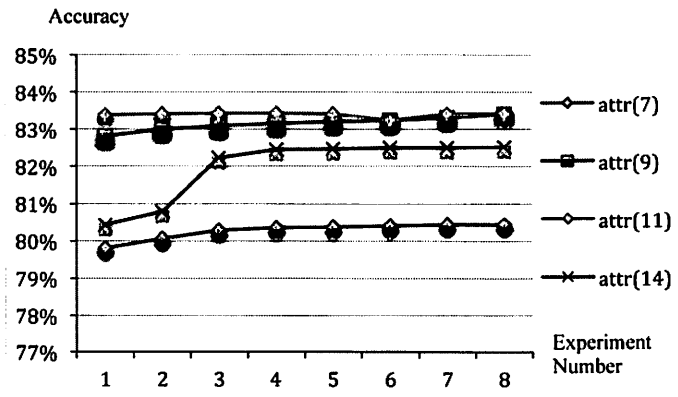

Fig. 4. Accuracy with different attribute number of training data

c) Analysis of relationship with attributes

With different number of attributes, the accuracy of percent obtained distinguishing levels. In the whole view, the accuracy increased while the number of attributes enhanced. But when the full attributes were used to do the experiment, it indicates that the accuracy and stability did not increase as expecting.

The reason of that situation was called Overfitting in statistics and machine learning area. Overfitting occurs when a statistical model describes random error or noise instead of the underlying relationship. Overfitting generally occurs when a model is excessively complex, such as having too many parameters relative to the number of observations. The possibility of overfitting exists because the criterion used for training the model is not the same as the criterion used to judge the efficacy of a model. In particular, a model is typically trained by maximizing its performance on some set of training data. However, its efficacy is determined not by its performance on the training data but by its ability to perform well on unseen data. Overfitting occurs when a model begins to memorize training data rather than learning to generalize from trend.

In the AROW thesis mentioned, there are comparisons with other algorithms. The result chart on that thesis indicates the online learning curves for both full and diagonalized versions of the algorithms on these noisy data. AROW improves over all competitors, and the full version outperforms the diagonal version. As the AROW introduced the synthetic data processed by several algorithms and the relationship between the number of instances and mistakes (10% noise, 100 runs). It improves the AROW algorithm owns the superior noise tolerance than other algorithms.

d) The accuracy with multiple processors in distributed environment

Machine learning cannot achieve 100% accuracy for classification. There are lots of factors, such as the complex of data, the high bias, low variance and etc. To guarantee the final result of predictions, a confidence on result of prediction is essential.

Assuming that there are N processors to do the prediction, every processor has the 80% accuracy to calculate the correct result (correct prediction). If over half of the results were the same, I could affirm that the result is reliable. In the case of the half against other half situation and could not distinguish the correct result. So the processors should have to do the prediction once again to avoid that kind of undefined situation. If the half-and-half situation cannot change after several reiterations, to kick out one processing result from clusters randomly to avoid the half-and-half situation. Below I defined the determination formulas based on Binomial distribution.

$$P(N) = \sum^{i} \binom{i}{N} (1 - P_c)^{N-i} (P_c)^{i} \quad \left(\frac{N}{2} < i \leq N\right) \tag{3}$$

$$P(N) = \sum_{i}^{i} \binom{i}{N} (1-P_c)^{N-i}(P_c)^i + \binom{\frac{N}{2}}{N} (1-P_c)^{\frac{N}{2}}(P_c)^{\frac{N}{2}} P(N-1) \quad \left(\frac{N}{2} < i \le N\right) \tag{4}$$

If N was an uneven number, implement the first formula to calculate the accuracy. If not, implement the second one and the accuracy on (N-1). The formulas indicate that the probability of the N processors could give the relationship between the accuracy probability and processors number. The expecting accuracy is based on the theoretical assumption of individual processors could achieve the accuracy of P_c.

According to the Table.2, just assume the P_c= 80% for AROW algorithm for processing NLP issue. Than the theoretical final accuracy on multiple processors would be calculated easily.

P_number	1	2	3	4
Accuracy (%)	80.00	89.60	89.60	95.58
5	6	7	8	9
94.21	97.83	96.67	98.81	98.06

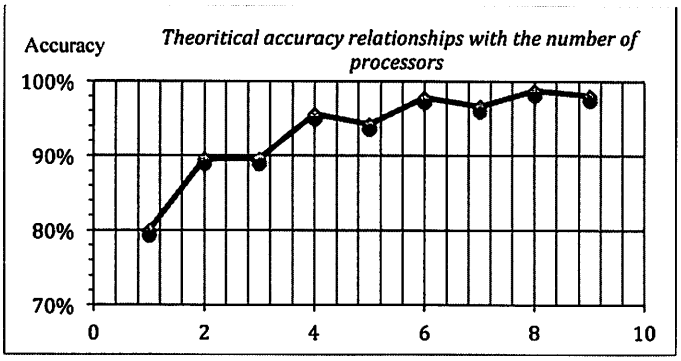

Fig. 5. Theoretical accuracy relationships with the nubmer processors

Based on the theoretical final accuracy drawing the Fig.7 above. The overall trend of accuracy is upgraded with the number of processors; we can obviously recognize that the formulas function of P (N) possess increasing feature and Infinite approach to 100%. The less fault rate happens, the better performance system achieves.

e) Training data set

If defining the training dataset for each processor contains X instances, the number from each classification is X*P (x), P (x) is that main attributes occupancy on original dataset. The instances of two training datasets with the some attributes should be distinguishing, so that the whole system would gain more dissimilar instances for training.

V. DEVELOPMENT OF EXPERIMENT ENVIRONMENT

A. Essential issues

The requirement software libraries list like table.2 below.

TABLE.2. THE ESSENTIAL SOFTWARE LIBRARIES FOR JUBATUS

Software	Version	Mandatory	Note
msgpack	>= 0.5.7	Yes	
jubatus-mpio	0.4.1	Yes	
jubatus-msgpack-rpc	0.4.1	Yes	C++ client library must be installed.
pficommon	master	Yes	Python (2.4 ~3.2) ,gcc-c++ (3.4 ~4.6)
google-glog	>= 0.3.3	Yes	
re2	master	No	
zookeeper	>= 3.3	No	C client library must be installed.
gcc	>=4.4	Yes	
pkg-config	>=0.26	Yes	
Python	>=2.4	Yes	
autoconf	>=2.59	Yes	Libraries installation tools

B. The flow of Jubatus development

- *Connection settings to jubaclassifier*
- *Prepare the training data*
- *Data training (update the model)*
- *Prepare the test data*
- *Classify the test data*
- *Output the result*

C. Single mode

1) Jubatus framework install

Simple install command:

$ ./waf configure $ sudo ./waf build $ sudo ./waf install

2) Jubatus clients

$ sudo pip install jubatus Or download the resource package *$ sudo python setup.py*

D. Distributed mode

1) Apache Zookeeper installing

The distributed mode install sequence is a little different from single mode. The ZooKeeper client libraries come in two languages: Java and C. In Jubatus framework, the Zookeeper should be installed at first with C client.

2) Jubatus frameworks and Jubatus clients

The frameworks and clients installation is similar to the single mode. But in distributed mode the Zookeeper should be enable status in Jubatus framework (default configure is unable). So when configure the install in Jubatus should implement the zookeeper library.

- *Start the Zookeeper server*
- *Register configuration file to ZooKeeper*
- *Configure Jubatus keeper to ZooKeeper*
- *Start Jubatus classifier agent process*
- *Start the distributed system*

VI. EXPERIMENT RESULT

A. The information of experiment issues

1) Hardware information of servers: Memory storage: 4G ; CPU: Inter Core2 CPU 6600@2.40GHZ*2; Hard Disk: 155.5G

2) Software information: Operation system: Ubuntu 12.04 LTS x64; Distributed framework: Jubatus 0.4.4, Zookeeper

3.4.5; Compiler: GCC 4.6; Programming language: Python; Algorithm: Adaptive Regularization of Weight Vector

3) Test data information: Size of data: 41023 records; Dimensions of records (attributes): 10; Type of records: text and integer; Processors: 6

B. Results of experiment

I have tested the part of the data from the original data of Muti-Domain Dataset in eight times. And recorded the result as below.

Test	1	2	3	4
Time(s)	11.72	11.64	11.95	11.76
5	6	7	8	Average
11.83	11.85	11.55	11.70	11.75

The table.4 presents the accuracy information of 6 processors accuracies and the final prediction accuracy in 8 tests.

TABLE.3 THE ACCURACY RESULTS (UNIT %)

Test	1	2	3	4
Pocessor1	76.54	76.33	75.59	76.52
Pocessor2	75.63	73.49	73.32	74.45
Pocessor3	73.73	73.52	74.45	75.21
Pocessor4	75.21	77.05	75.92	75.35
Pocessor5	76.11	76.32	75.87	76.15
Pocessor6	76.52	76.23	76.32	75.98
Final_Result	90.23	91.12	89.75	90.12
5	6	7	8	Average
77.12	76.87	75.62	77.21	76.48
75.34	75.34	76.65	75.35	74.95
73.87	75.10	74.54	74.43	74.36
77.12	75.89	75.92	75.33	75.72
76.21	75.98	76.13	76.22	76.11
76.21	75.87	75.54	76.31	76.12
91.01	90.31	90.50	90.45	90.44

For one processor, the accuracy is between 74%~77% and lower than the single mode average accuracy 83% mentioned above.

According to the formula (3), (4) and assume the theoretical accuracy of one processor is 75%; the accuracy should be 92.88% with 6 processors. But the average final accuracy in distributed mode with parallel computing of processors is lower than the theoretical value.

The average rate of processing data could be calculated by 41023/11.75=3491 queries/s. that is corresponding to the theoretical rate in the introduction of Jubatus documents.

The dataset complexity: The precision and performance of machine learning are strongly dependent on the dataset. The dataset complexity and dimension would bring a significant effect on accuracy. To optimize and regularize the training data and input arguments is an efficient way to improve the performance of data processing.

Process loss in distributed mode in Jubatus: The Jubatus document described that experiments in a distributed environment with eight nodes utilizing the mix demonstrated that there was an overall loss of 12.7% of the training data. The mix is the key method of Jubatus to perform asynchronously with respect to the training of the entire system by an iterative parameter mixture in online classification and online regression problems in a distributed environment. This characteristic will be useful when statistical machine learning is applied to large quantities of data, because the number of items of training data per unit time increased in a substantially linear manner by implementing the mix method.

Comparing with experiments in other thesis, the performance in my experiment is satisfactory. It has outstanding ability of enormous query processing and lower resource cost than normal big data searching. But the accuracy still needs to be improved for high accuracy requirement.

VII. CONCLUTIONS AND FUTURE WORK

The design for big data searching, improve the efficiency while the accuracy within the receivable extend. The purpose of the project attempts to combine machine learning and parallel computing in big data searching. The project basically finished, but it still exists some shortage. The generalization for other dataset still to be developed and the accuracy need to be promoted. The accuracy also has a major relationship with the algorithms. The distributed system and parallel computing cooperation optimization issues when the cluster environment exist different performance processors. The project is just started and it is rather early to evaluate that kind of proposed cooperation and design. But I believe the machine learning technology will develop more and more mature and do a great devotion to the big data searching in the future.

The remaining issues are how to improve the single processing accuracy in distributed mode, and the false result processing. I will focus on that remaining issues and continue to improve and optimize it.

ACKNOWLEDGEMENT

I would like to thank Professor Nobuhiko Koike, for his proposals and help. Moreover, thank for the people, who help me to a solution and approach. And about the Jubatus issues, I appreciate the support of the developers of Jubatus online. Without the help from them, the research cannot be realized.

REFERENCE

[1] Satoshi Oda, Kota Uenishi, and Shingo Kinoshita, Jubatus: Scalable Distributed Processing Framework for Realtime Analysis of Big Data, Nippon Telegraph & Telephone Corp, Preferred Infrastructure Corp. October 26, 2011.

[2] Apache Zookeeper. http://zookeeper.apache.org/

[3] David J. C. MacKay. Information Theory, Inference, and Learning Algorithms Cambridge: Cambridge University Press, 2003. ISBN 0-521-64298-1

[4] Martin Farach Colton, http://www.tokutek.com. December 11,2012

[5] Rosenblatt, Frank (1957), The Perceptron--a perceiving and recognizing automaton. Report 85-460-1, Cornell Aeronautical Laboratory.

[6] Crammer K, Dekel O, Keshet J, et al. Online passive-aggressive algorithms [J]. The Journal of Machine Learning Research, 2006, 7: 551-585.

[7] Mark Dredze, Koby Crammer and Fernando Pereira, Confidence-Weighted Linear Classification, Proceedings of the 25th International Conference on Machine Learning (ICML), 2008

[8] Koby Crammer, Alex Kulesza and Mark Dredze, Adaptive Regularization Of Weight Vectors, Advances in Neural Information Processing Systems, 2009

[9] Koby Crammer and Daniel D. Lee, Learning via Gaussian Herding, Neural Information Processing Systems (NIPS), 2010

[10] Hamilton Institute. "The Binomial Distribution" October 20, 2010