www.ijecs.in

International Journal Of Engineering And Computer Science ISSN:2319-7242
Volume 4 Issue 1 January 2015, Page No. 9971-9976

Predicting Student Performance Using MapReduce

Dr. N. Tajunisha[1], *M. Anjali*[2]

[1]Associate Professor
Department of Computer Science
Sri Ramakrishna college of Arts and Science for women, Coimbatore
tajkani@gmail.com

[2]Research Scholar
Department of Computer Science
Sri Ramakrishna college of Arts and Science for women, Coimbatore
anjalivivekjc@gmail.com

Abstract: ***Data mining and machine learning depend on classification which is the most essential and important task. Many experiments are performed on Student datasets using multiple classifiers and feature selection techniques. Many of them show good classification accuracy. The existing work proposes to apply data mining techniques to predict Students dropout and failure. But this work doesn't support the huge amount of data. It also takes more time to complete the classification process. So the time complexity is high. To improve the accuracy and reduce the time complexity, the MapReduce concept is introduced. In this work, the deadline constraint is also introduced. Based on this, an extensional MapReduce Task Scheduling algorithm for Deadline constraints (MTSD) is proposed. It allows user to specify a job's (classification process in data mining) deadline and tries to make the job to be finished before the deadline. Finally, the proposed system has higher classification accuracy even in the big data and it also reduced the time complexity.***

Keywords: Data Mining, Big Data, Classification, MapReduce.

1. Introduction

The main objective of higher education institutes is to provide quality education to its student and to improve the quality of managerial decisions. One way to achieve highest level of quality in higher education system is by discovering knowledge from educational data to study the main attributes that may affect the student's performance. The discovered knowledge can be used to offer a helpful and constructive recommendations to the academic planners in higher education institutes to enhance their decision making process, to improve student's academic performance and trim down failure rate, to better understand student's behaviour, to assist instructors, to improve teaching and many other benefits.

Educational data mining [25] uses many techniques such as decision tree, rule induction, neural networks, k-nearest neighbour, naïve Bayesian and many others. By using these techniques, many kinds of knowledge can be discovered such as association rules, classifications and clustering. It showed what kind of data could be collected, how to pre-process the data, how to apply data mining methods on the data, and finally how to get benefited from the discovered knowledge. Many kinds of knowledge can be discovered from the extracted data.

In this era of big data, huge amounts of structured and unstructured student data are being produced daily. Big Data is difficult to work with and requires massively parallel software running on a large number of computers. Previous works investigated the most common ones which are attribute selection, classification such as decision tree method. The preliminary work proposes to apply data mining techniques to predict student's dropout and failure. But this work doesn't support the huge amount of data. And also it takes more time to complete the classification process. So the time complexity is high. To improve the accuracy and reduce the time complexity, the MapReduce concept is introduced.

MapReduce is a recent programming model that simplifies distributed applications that handle Big Data. For implementing the concept of MapReduce, it has to divide the student data and perform the data mining process. After that the aggregated result is produced. Consequently, the performance of MapReduce strongly depends on how evenly it distributes the student data. In MapReduce [27], workload distribution depends on the algorithm that partitions the data. To improve the accuracy of the system and supports the big data, the novel technique is proposed in this work. In this work, the deadline constraint is also introduced. Based on this, an extensional MapReduce Task Scheduling algorithm for Deadline constraints (MTSD) is proposed. It allows user to specify data mining process's deadline and tries to make the data mining process to be finished before the deadline.

Increasing digitization of student records means predictive analytics is expected to transform teaching and become a key tool in learning more about students. Predictive analytics is a process in which data collected about the student, typically attendance, subjects taken, assessment is used to understand learning patterns, identify skill gaps, predict performance and identify learning opportunities. The effective feature selection method is required to analyze the efficient classification algorithm. Handling big data, a novel partition mechanism in MapReduce is also required.

2. Background Study

Angie Parker [1], says that the college offered the courses to students in both traditional and distance formats. In the semester the traditional classes has less than 3 percent dropout rate and distance classes had exceeded 17 percent dropout rate. The author proposed a Locus of Control Scale to determine the variable combinations that could be considered as predictors of dropouts from distance education. Finally the variable was reserved for inclusion in the discriminant analysis the result was reexamined.

Haun Liu and Dan Orban et al [13], says that the cloud OS is responsible for managing resources. Cloud OS is responsible for managing the cloud infrastructure, hiding the cloud infrastructure details from the application programmers and coordinating the sharing of limited resources. The author describes how cloud Mapreduce is implemented using Amazon cloud OS. The MapReduce framework collects the output key value pairs from the map function and writes them to reduce the queues.

Jeffrey Dean and Sanjay Ghemawat et al [16], suggest that MapReduce is a programming model and an associated implementation for processing and generating large data sets. User specify a map function that processes a key /value pairs and a reduce function that merges all intermediate values associated with same intermediate key. MapReduce runs on a large cluster of commodity machines and highly scalable. The author proposed a major contribution to this work, which are simple and powerful interface that enables automatic parallelization and distribution of large scale computation, combined with an implementation of this interface achieves high performance on large clusters of commodity PCs.

Quadri M. N. and Kalyankar N. V. et al [21], says that in predictive modeling, the decision tree techniques provide decision making process. The process starts by studying the relationship between the student dropout and the risk factors including gender, attendance, scholarship, parent income and whether the student is working or not. The author developed a hybrid method for this study. As a result the decision tree algorithm is used to identify the factors that influence dropouts. After the factors are identified, the logistic regression technique is used to quantify the dropouts and effect of each risk factor.

MateiZaharia and Anthony D. Joseph et al [19], says that Hadoop is an open source implementation of MapReduce enjoying wide adoption and used for short jobs where low response time is critical. In this research the author proposed improved scheduling algorithm that reduces Hadoops response time and addresses the problem of how to robustly perform execution to maximize the performance.

Many authors have applied data mining techniques to predict student performance. But it doesn’t support huge amount of data. So the Mapreduce concept is proposed in this work to handle large data and improve it accuracy and time complexity.

3. Proposed Methodology

This work proposes an extensional MapReduce Task Scheduling algorithm for Deadline constraints (MTSD). It allows user to specify a classification process in data mining deadline and tries to make the data mining process be finished before the deadline. This algorithm classifies the student data into several levels. Under this algorithm, first it illuminates a novel data distribution model which distributes student data according to the student’s data capacity level respectively. The experiments show that the student data classification algorithm can improve data locality observably to compare with default scheduler and it can also improve other scheduler’s locality. Secondly, it calculates the data mining process’s average completion time which is based on the student data level. It improves the precision of classification’s remaining time evaluation.

The MTSD algorithm takes the student data locality and cluster heterogeneity into account. The data locality is the key factor that affects the efficiency of MapReduce classification process. The data locality means that the classification’s operation code and the classification’s input data are on the same computing node or on the same rack. Of course, the efficiency when the code and data are on the same node is higher than on the same rack. If the code and data are on the same node, it would avoid the data transmission in the network and greatly reduce the delay. Therefore, in the large scale data processing applications, shifting the code would be “cheaper” than moving data. In this work, in order to meet the time constraints of the data mining process and further improve the data locality, the MapReduce Task Scheduler for Deadline (MTSD) algorithm is proposed, which is based on the computing power of meet time constraints in the heterogeneous environments.

Methodology

Step 1: Data gathering from school.
Step 2: Data preprocessing task is carried out.
Step 3: In MapReduce concept, Mapper partitions the student data with key values, after that Reducer aggregates the result using MTSD algorithm. Consequently, the performance of MapReduce strongly depends on how evenly it distributes the student data.
Step 4: Attribute selection takes place in different terms such as Correlation-based Feature Selection (CFS).
Step 5: Classification is done using decision tree and rule induction.
Step 6: Imbalanced dataset classification is done with SMOTE algorithm.
Step 7: Finally performance is evaluated using several terms such as True Positive Rate (TPR), False Positive Rate (FPR), False Negative Rate (FNR), True Negative Rate (TNR), Accuracy and Time Comparison.

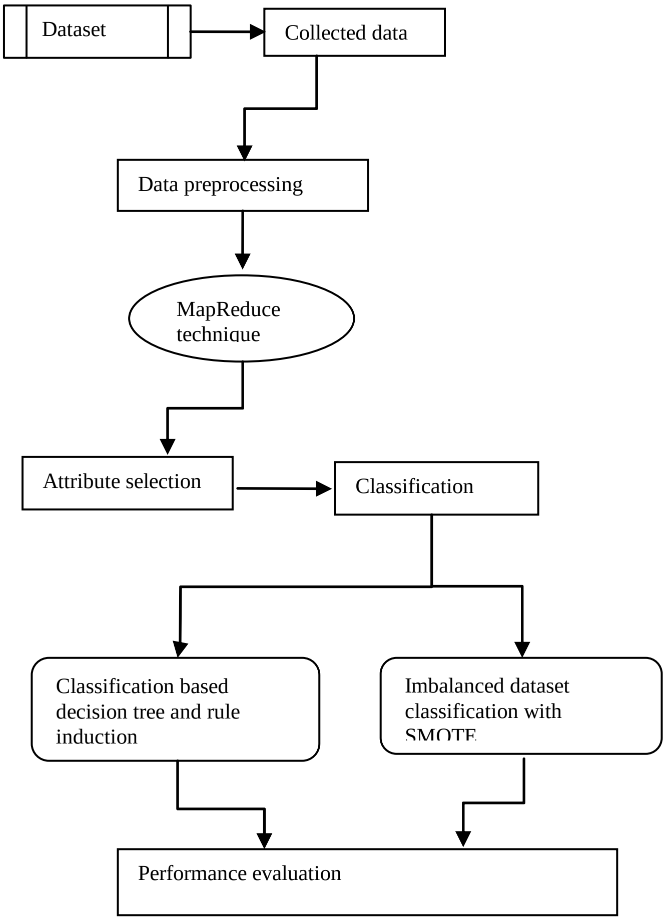

Figure 1: Flow diagram of proposed method

3.1 Data gathering from school

All the information used in this study has been gathered from different sources. In this research work synthetic dataset has been used and a specific survey was designed to administer all students in the middle of the course. Its purpose was to obtain personal and family information to identify some important factors that could affect school performance. From a general survey [14] which is completed when the students register in the National Evaluation Center (CENEVAL) for admission to many institutions of secondary and higher education.

3.2 Data preprocessing

Before applying DM algorithm it is necessary to carry out some pre-processing tasks such as cleaning, integration, discretization and variable transformation. It must be pointed out that very important task in this work was data pre-processing, due to the quality and reliability of available information, which directly affects the results obtained. In fact, some specific pre-processing tasks were applied to prepare all the previously described data so that the classification task could be carried out correctly. First, all available data were integrated into a single dataset. During this process inefficient student's information are eliminated. Some modifications are also made to the values of some attributes. Furthermore, the continuous variables are transformed into discrete variables, which provide a much more comprehensible view of the data.

3.3 MapReduce

In the proposed system the performance of the system is improved by using MapReduce. This is simple yet powerful framework which lets the programmer write simple units of work as **map** and **reduce** functions. In summary, they are:

- "In-mapper combining", where the functionality of the combiner is moved into the mapper. Instead of emitting intermediate output for every input key-value pair, the mapper aggregates partial results across multiple input records and only emits intermediate key-value pairs after some amount of local aggregation is performed.

- The related patterns "pairs" and "stripes" for keeping track of joint events from a large number of observations. The pairs approach keeps track of each joint event separately, whereas the stripes approach keeps track of all events that co-occur with the same event. Although the stripes approach is significantly more efficient, it requires memory on the order of the size of the event space, which presents a scalability bottleneck.

- "Order inversion", where the main idea is to convert the sequencing of computations into a sorting problem. Through careful orchestration, one can send reducer the result of a computation (e.g., an aggregate statistic) before it encounters the data necessary to produce that computation.

- "Value-to-key conversion", which provides a scalable solution for secondary sorting. By moving part of the value into the key, one can exploit the MapReduce execution framework itself for sorting.

Based on MapReduce concept, an extensional MapReduce Task Scheduling algorithm for Deadline constraints (MTSD) is proposed. It allows user to specify data mining process's deadline and tries to make the data mining process to be finished before the deadline.

3.4 Attribute selection

The attributes are selected using the Feature Selection Techniques [18] called, Correlation-based Feature Selection (CFS).

The Correlation-based Feature Selection (CFS) estimates and ranks the subset of features than individual features. It chooses the set of attributes that are highly associated with the class, in addition to those attributes that are in low inter-correlation.

3.5 Decision tree classification

A decision tree is a simple structure where non-terminal nodes represent tests on one or more attributes and terminal nodes reflect decision outcomes. The information gain measure is used to select the test attribute at each node of the decision tree. The information gain measure prefers to select attributes having a large number of values. The basic decision tree induction algorithm ID3 was enhanced by C4.5. C4.5 a successor of ID3 uses an extension of information gain known as gain ratio, which attempts to overcome this bias. The WEKA classifier package has its own version of C4.5 known as J4.8. J4.8 is used to identify the significant attributes.

3.6 Imbalanced dataset classification with Smote

The problem of imbalanced data classification occurs when the number of instances in one class is much smaller than the number of instances in another class or other classes. Traditional classification algorithms have been developed to maximize the overall accuracy rate, which is independent of class distribution. This means that the majority of class classifiers are in the training stage, which leads to low sensitivity classification of minority class elements at the test stage. One way to solve this problem is to act during the pre-processing of data by carrying out a sampling or balancing of class distribution. There are several data balancing or rebalancing algorithms; one that is widely used and that is available in Weka as a supervised data filter is SMOTE (Synthetic Minority Oversampling Technique). In the SMOTE algorithm, the minority class is over-sampled by taking each minority class sample and introducing synthetic examples along the line segments joining any or all of the *k* minority class nearest neighbors. Depending on the amount of over-sampling required, neighbors from the *k* nearest neighbors are randomly chosen.

3.7 Performance Evaluation

Finally, the performance of the existing classification and prediction system is measured with proposed grammar based genetic programming approach to derive the pass/failure result. Measure the performance results in terms of the true positive rate (TPR), false positive rate (FPR), False Negative Rate (FNR) and True negative Rate (TNR), accuracy, Time comparison.

4. Result Analysis

The experiments show that the student data classification algorithm can improve data locality observably to compare with default scheduler and it also can improve other scheduler's locality. Secondly, calculate the data mining process's average completion time which is based on the student data level. It improves the precision of classification's remaining time evaluation. Finally the result shows that the proposed system has higher classification accuracy in big data and also reduces the time complexity.

Results and Discussion:

Here analyze and compare the performance offered classification without MapReduce technique and MapReduce based classification. The performance is evaluated by the parameters such as accuracy and time complexity rate. Based on the comparison and the results from the experiment, the proposed approach works better than the existing system.

Accuracy Rate:

Accuracy is defined as the overall accuracy rate or classification accuracy and is calculated as

$$\text{Accuracy} = \frac{\text{True positive} + \text{True}}{\text{True positive} + \text{True negative} + \text{False}}$$

- **TP (True positive)**

In a statistical hypothesis test, there are two types of incorrect conclusions that can be drawn. The hypothesis can be inappropriate. A positive test results accurately reflects the test for activity and to analyze. If the outcome from a prediction is p and the actual value is also p, then it is called a true positive (TP);
True positive rate (TPR) =TP/P
P= (TP+FN)
Where P is positive, TP is the True Positive

- **TN (True negative)**

A true negative (TN) has occurred when both the prediction outcome and the actual value are n in the number of input data.
True negative rate (TNR) =TN/N
N= (TN+FN)
Where
N is the Negative value.
TN is the True Negative.

- **FP (False positive)**

If the outcome from a prediction is p and the actual value is n then it is said to be a false positive (FP).
False positive rate (α) = FP / (FP + TN)

- **FN (False negative)**

False negative (FN) is when the prediction outcome is n while the actual value is p.
False negative rate (β) =FN / (TP + FN)

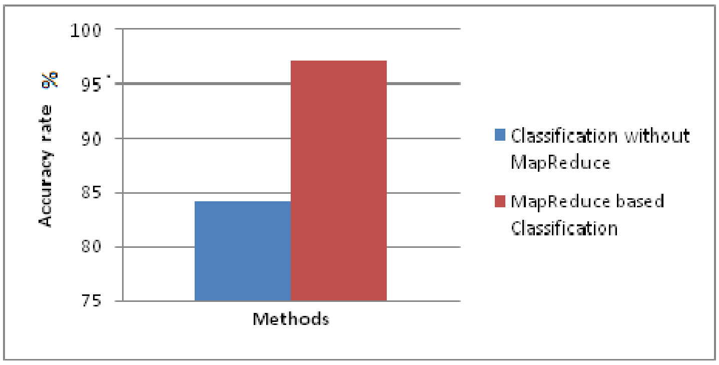

Figure: 2 Accuracy comparisons

This graph shows the accuracy rate of existing classification without MapReduce technique and proposed system such that MapReduce based classification based on two parameters of accuracy and methods such as existing and proposed system. From the graph we can see that, accuracy of the system is reduced somewhat in existing system than the proposed system. From this graph we can say that the accuracy of proposed system is increased which will be the best one.

Time Complexity

The proposed system is evaluated in terms of the time complexity. In other words computation time of the classification technique with the existing classification without MapReduce technique and proposed system such that MapReduce based classification. It is defined as the time taking for the classification process in whole data in the existing system compared with proposed system such that MapReduce based classification.

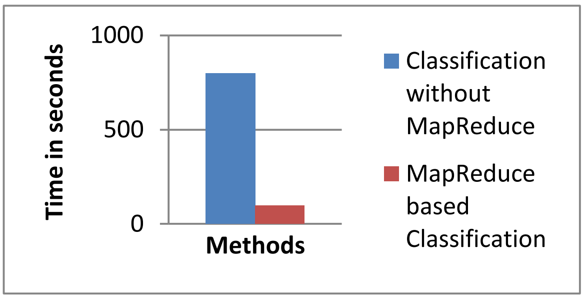

Figure: 3 Time complexity comparisons

In the above graph, we are comparing the time complexity of the existing classification without MapReduce technique and proposed system such that MapReduce based classification. In this graph, x axis will be the methods (existing and proposed system) and y axis will be time in seconds. From the graph we can easily understand that the proposed system has very low time. So the proposed system has low time complexity compared to the existing system.

5. Conclusion and Future work

Conclusion:

The aim of this study is to analyze factors affecting academic achievement that contribute to the prediction of students' academic performance. It is useful in identifying weak students who are likely to perform poor in their studies. An educational institution needs to have an approximate prior knowledge of enrolled students to predict their performance in future academics. The various data mining techniques can be effectively implemented on educational data. From the above results it is clear that classification techniques and mapreduce concept can be applied on educational data for predicting the student's outcome and improves their results. The classification accuracy and performance is high in the proposed system. This experiment shows that the proposed system is more efficient than the existing system. The advantages of this system are

- ➢ Time complexity of the system is reduced
- ➢ It supports for big data. So it will be applicable in several real applications. It also supports the online data.
- ➢ The classification accuracy is increased even for big data.
- ➢ Supports the dataset with redundant data and also the computational complexity is low compared to the existing system
- ➢ The efficiency of the system is improved.

Finally, the next step in our research is to carry out experiments using more data and also from different educational levels (primary, secondary, and higher) to test whether the same performance results are obtained using different DM approaches. As future work, we can mention the following:

- To predict the student failure as soon as possible. The earlier the better, in order to detect students at risk in time before it is too late.
- To propose actions for helping students identified within the risk group. Then, to check the rate of the times it is possible to prevent the fail or dropout of that student previously detected.

References

[1] Angie Parker, "A study of variables that predict dropout from distance education," Int. J. Educ. Technol., vol. 1, no. 2, pp. 1–11, 1999.

[2] Araque F, C. Roldán, and A. Salguero, "Factors influencing university dropout rates," *Comput. Educ.*, vol. 53, no. 3, pp. 563–574, 2009.

[3] Breiman L, J. H. Friedman, R. A. Olshen, and C. J. Stone, *Classification and Regression Trees*. New York, USA: Chapman & Hall, 1984.

[4] Cendrowska J, "PRISM: An algorithm for inducing modular rules," *Int. J. Man-Mach. Stud.*, vol. 27, no. 4, pp. 349–370, 1987.

[5] Chawla N. V. and K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "Synthetic minority over-sampling technique," *J. Artif. Intell. Res.*, vol. 16, pp. 321–357, Jun. 2002.

[6] Chao Jin and Christian Vecchiolaand RajkumarBuyya , "MRPGA: An Extension of MapReduce for Parallelizing Genetic Algorithms", Grid Computing and Distributed Systems (GRIDS) Laboratory, Department of Computer Science and Software Engineering , The University of Melbourne, Australia , 2008.

[7] Chris Miceli and Michael Miceli, Bety Rodriguez-Milla and ShantenuJha, "Understanding performance of distributed data-intensive applications", 2009.

[8] Cristobal Romero and Sebastian Ventura, "Educational data mining: A survey from 1995 to 2005," *Expert Syst. Appl.*, vol. 33, no. 1, pp. 135–146, 2007.

[9] Espíndola E and A. León, "La deserción escolar en américalatina: Un Temaprioritario para la agenda regional," *RevistaIberoamer. Educ.*, vol. 1, no. 30, pp. 39–62, 2002.

[10] Fay Chang, Jeffrey Dean, Sanjay Ghemawat, Wilson C. Hsieh, Deborah A. Wallach Mike Burrows, Tushar Chandra, Andrew Fikes, Robert E. Gruber, "Bigtable: A Distributed Storage System for Structured Data" 2006.

[11] Freund Y and L. Mason, "The alternating decision tree algorithm," in *Proc. 16th Int. Conf. Mach. Learn.*, 1999, pp. 124–133.

[12] Fourtin L, D. Marcotte, P. Potvin, E. Roger, and J. Joly, "Typology of students at risk of dropping out of school: Description by personal, family and school factors," *Eur. J. Psychol. Educ.*, vol. 21, no. 4, pp. 363–383, 2006.

[13] Huan Liu, Dan Orban, "Cloud MapReduce: a MapReduce Implementation on top of a Cloud Operating System", Accenture Technology Labs, IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing, 2011.

[14] Hernández M. M, "Causas del fracaso escolar," in Proc. 13th Congr. Soc. Española Med. Adolescente, 2002, pp. 1–5.

[15] Ian H. Witten and F. Eibe, *Data Mining, Practical Machine Learning Tools and Techniques*, 2nd ed. San Mateo, CA, USA: Morgan Kaufman, 2005.

[16] Jeffrey Dean., Ghemawat, S.: MapReduce: simplified data processing on large clusters. Commun. ACM 51(1), 107–113 (2008). doi:10.1145/1327452.1327492

[17] Lykourentzou I, I. Giannoukos, V. Nikolopoulos, G. Mpardis, and V. Loumos, "Dropout prediction in e-learning courses through the combination of machine learning techniques," *Comput. Educ.*, vol. 53, no. 3, pp. 950–965, 2009.

[18] Mark A. Hall and G. Holmes, "Benchmarking attribute selection techniques for data mining," Dept. Comput. Sci., Univ. Waikato, Hamilton, New Zealand, Tech. Rep. 00/10, Jul. 2002.

[19] MateiZaharia, Konwinski, A., Joseph, A.D., Katz, R., Stoica, I.: Improving MapReduce performance in heterogeneous environments. In: Proceedings of the 8th USENIX Conference on Operating Sys- tems Design and Implementation, pp. 29–42 (2008)

[20] Moseley L. G and D. M. Mead, "Predicting who will drop out of nursing courses: A machine learning exercise," *Nurse Educ. Today*, vol. 28, no. 4, pp. 469–475, 2008.

[21] Quadri M. N and N. V. Kalyankar, "Drop out feature of student data for academic performance using decision tree techniques," *Global J.Comput. Sci. Technol.*, vol. 10, pp. 2–5, Feb. 2010.

[22] Quinlan J. R, *C4.5: Programs for Machine Learning*. San Mateo, CA, USA: Morgan Kaufman, 1993.

[23] Sanjay Ghemawat, Howard Gobioff, and Shun-Tak Leung, “The Google File System”, 2003.

[24] SelcukCandan K, RanKloud: “Scalable Multimedia and Social Media Retrieval and Analysis in the Cloud”, Arizona State University, 2011.

[25] Sotiris Kotsiantis, “Educational data mining: A case study for predicting dropout—prone students,” *Int. J. Know. Eng. Soft Data Paradigms*, vol. 1, no. 2, pp. 101–111, 2009.

[26] Steffen Heinz, Justin Zobel, Hugh E. Williams, “Burst Tries: A Fast, Efficient Data Structure for String Keys”, 2002.

[27] Sven Groot, Masaru Kitsuregawa, “Jumbo: Beyond MapReduce for Workload Balancing”,Institute of Industrial Science, The University of Tokyo, 4-6-1 Komaba Meguro-ku, Tokyo 153-8505, Japan, 2010.