The Journal of Systems and Software 122 (2016) 83–92

Contents lists available at ScienceDirect

The Journal of Systems and Software

journal homepage: www.elsevier.com/locate/jss

Big data mining with parallel computing: A comparison of distributed and MapReduce methodologies

Chih-Fong Tsai[a,*], Wei-Chao Lin[b], Shih-Wen Ke[c]

[a] Department of Information Management, National Central University, Taiwan
[b] Department of Computer Science and Information Engineering, Asia University, Taiwan
[c] Department of Information and Computer Engineering, Chung Yuan Christian University, Taiwan

ARTICLE INFO

Article history:
Received 3 November 2015
Revised 21 April 2016
Accepted 3 September 2016
Available online 5 September 2016

Keywords:
Big data
Data mining
Parallel computing
Distributed
Cloud computing
MapReduce

ABSTRACT

Mining with big data or big data mining has become an active research area. It is very difficult using current methodologies and data mining software tools for a single personal computer to efficiently deal with very large datasets. The parallel and cloud computing platforms are considered a better solution for big data mining. The concept of parallel computing is based on dividing a large problem into smaller ones and each of them is carried out by one single processor individually. In addition, these processes are performed concurrently in a distributed and parallel manner. There are two common methodologies used to tackle the big data problem. The first one is the distributed procedure based on the data parallelism paradigm, where a given big dataset can be manually divided into n subsets, and n algorithms are respectively executed for the corresponding n subsets. The final result can be obtained from a combination of the outputs produced by the n algorithms. The second one is the MapReduce based procedure under the cloud computing platform. This procedure is composed of the map and reduce processes, in which the former performs filtering and sorting and the later performs a summary operation in order to produce the final result. In this paper, we aim to compare the performance differences between the distributed and MapReduce methodologies over large scale datasets in terms of mining accuracy and efficiency. The experiments are based on four large scale datasets, which are used for the data classification problems. The results show that the classification performances of the MapReduce based procedure are very stable no matter how many computer nodes are used, better than the baseline single machine and distributed procedures except for the class imbalance dataset. In addition, the MapReduce procedure requires the least computational cost to process these big datasets.

1. Introduction

As a consequence of the popularity and advancement of related web and information technology, massive amounts of data are produced in our daily life. Large volumes of information, petabytes of data, are recorded every day. Clearly, the era of big data has arrived (Mayer-Schonberger and Cukier, 2014). In addition to the data size (i.e. volume), big data has other characteristics, such as variety and velocity. The former means that big data can be composed of a wide variety of structured, semi-structured, and unstructured data whereas the latter refers to the requirement of real-time processing and analysis (Fernandez et al., 2014). As a result, big data analytics by machine learning and data mining techniques has become an important research problem (Rajaraman and Ullman, 2011; Wu et al., 2014; Zhou et al., 2014).

Mining with big data or big data mining is very hard to manage using the current methodologies and data mining software tools due to their large size and complexity (Fan and Bifet, 2012). In other words, using a single personal computer (PC) to execute the data mining task over large scale datasets requires very high computational costs. It is necessary to use more powerful computing environments to efficiently process and analyze big data.

According to Wu et al. (2014), the solutions for the problem of mining large scale datasets can be based on the parallel and cloud computing platforms. In principle, parallel computing focuses on dividing the chosen (large) problem into smaller ones, each of which (i.e. calculation) is carried out by one single processor individually, so that a computation composed of a number of calculations is performed concurrently in a distributed and parallel manner (Gottlieb and Almasi, 1989). This leads to some research issues

* Corresponding author. Fax: +886 3 4254604.
E-mail address: cftsai@mgt.ncu.edu.tw (C.-F. Tsai).

http://dx.doi.org/10.1016/j.jss.2016.09.007
0164-1212/

for distributed data mining (Zheng et al., 2012) and distributed machine learning (Peteiro-Barral and Guijarro-Berdinas, 2013).

Specifically, from the data point of view, the data parallelism paradigm, called the distributed methodology in this paper, can be considered for processing large scale datasets. In data parallelism, the large scale dataset is partitioned among a number of processors, each of which executes the same computation (or mining algorithm) over a designated partition (or subset) (Zaki, 2000).

In the literature, the idea of the distributed methodology has been employed in ensemble classifiers. That is, each classifier is trained by a portion of a given training set. Then, to classify a new unknown test case, the test case is input into all of the trained classifiers, which make these classifiers work in a distributed and parallel manner. Finally, the classification results produced by these classifiers are combined via some combination methods, such as voting and weighted voting, for the final output (Kittler et al., 1998).

Recently, cloud computing has further extended the parallelism principle to effectively manage the utility and consumption of computing resources over a computer cluster. To handle large scale dataset problems, the MapReduce computation is usually implemented using Hadoop[1], a powerful parallel programing framework (Dean and Ghemawat, 2010). The MapReduce methodology is composed of the map and reduce procedures, in which the former performs filtering and sorting and the latter is a summary operation in order to produce the final result. There are many related studies focusing on this methodology for big data mining, such as attribute reduction (Qian et al., 2015), instance selection (Triguero et al., 2015), and class imbalance (Lopez et al., 2015).

According to above discussion, big data mining can be efficiently performed via the conventional distributed and MapReduce methodologies. Both methodologies require a number of processors (or computer nodes) to execute some mining tasks in a parallel manner. One difference between these two methodologies is the computing resource management. For the conventional distributed methodology, one can partition a large scale dataset into N subsets for N computer nodes to perform the mining task. In particular, each computer node can be managed manually, and each node is usually set to consume the same computing resources over each of the N subsets.

On the other hand, the MapReduce methodology automatically manages the consumption of the computing resources of different computer nodes when handling a large scale dataset. That is, there is no need to partition the large scale dataset into N subsets. Only the setting of the number of computer nodes to process the dataset, the map procedure, is required. However, each node does not necessarily consume the same computing resources.

This raises an important research question, which has never been asked before: Do the distributed and MapReduce methodologies perform differently over large scale datasets in terms of mining accuracy and efficiency? Therefore, the contribution of this study is to examine the mining performances of the distributed and MapReduce methodologies over large scale datasets in terms of classification accuracy and processing times. In addition, as using more computer nodes does not guarantee efficiency and accuracy, because of the related overheads, different numbers of computer nodes will be used for comparison.

The rest of this paper is organized as follows. Section 2 overviews the related literature for distributed and MapReduce-based data mining. Sections 3 and 4 present the experimental procedures and results, respectively. Finally, some conclusions are offered in Section 5.

[1] https://hadoop.apache.org/

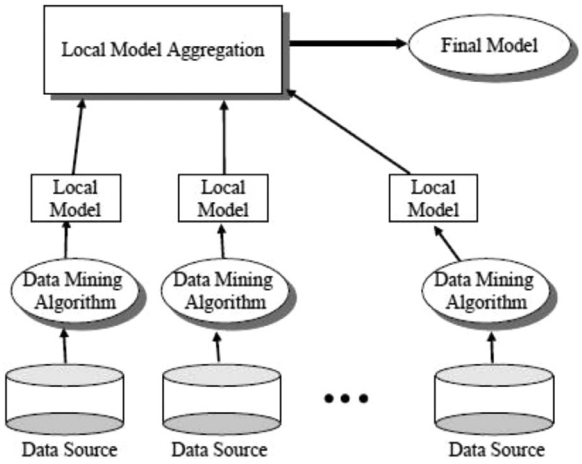

Fig. 1. Distributed data mining framework (Park and Kargupta, 2002).

2. Literature review

2.1. Distributed data mining

Distributed computing can refer to the use of distributed systems to solve computational problems. In particular, a problem is divided into many tasks, each of which is solved by one or more computers (or processors) that run concurrently in parallel. In addition, each processor can communicate with each other by message passing (Coulouris et al., 2011).

In the traditional data mining approach, the data are usually centralized and a specific algorithm is then chosen to process and analyze the data under a single computing platform. However, for a big data problem or large scale data mining, this is not so simple, and the performing the data mining tasks under the distributed computing platform has become an important area of research investigation (Zaki, 2000; Zheng et al., 2012).

Generally speaking, the objective of distributed data mining is to perform the data mining tasks based on the distributed resources, including the data, computers, and data mining algorithms (Park and Kargupta, 2002). Fig. 1 shows a general distributed data mining framework where different data sources may be homogenous and/or heterogeneous. Each data mining algorithm handles its corresponding data source under a single computing platform leading to a local model. Then, these local models are aggregated in order to generate the final model.

To solve the big data problems, the data parallelism paradigm can be considered. Given a large scale dataset D, it can be divided into n subsets, denoted as D_1, D_2, D_3, ...D_n, where each subset may contain different numbers of data samples and each subset may or may not have duplicate data samples. Then, a specific data mining algorithm implemented in n local machines (or computer nodes) individually is performed over each subset. Finally, the n mining results are combined via one combination component to produce the final output.

This distributed approach is different from the traditional one that performs the data mining algorithm over D directly on a single machine. As we can imagine, when the dataset size becomes very large, the processing time for the traditional approach greatly increases, but the distributed approach can tackle this large scale dataset problem in an efficient way.

In practice, most users usually only have limited computing resources to perform big data mining. One advantage of this approach is that only a single computer is used to tackle each subset at a time, and that machine performs the same task for different subsets n times, resulting in n different models. The different re-

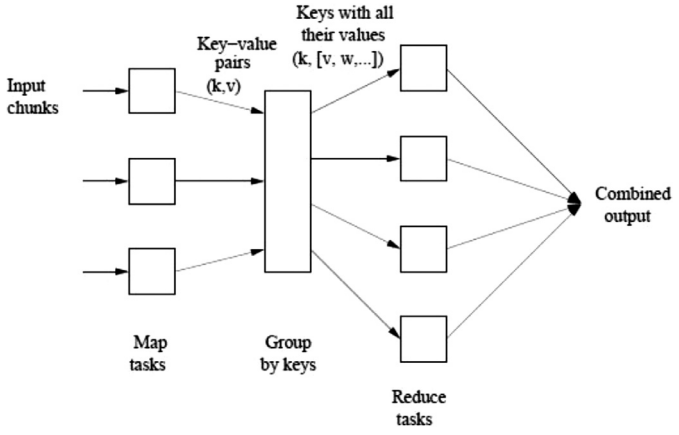

Fig. 2. MapReduce computation procedure (Rajaraman and Ullman, 2011).

sults obtained by these different models can then be combined for the final output.

However, one limitation of the distributed data mining approach to the big data problem is the need to manually partition the chosen data and allocate different computer nodes (or processors) to perform the mining task over the partitioned subsets.

2.2. MapReduce-based data mining

The MapReduce framework is implemented in the cloud computing environment. This can be regarded as a new generation of programming system for the parallel processing purposes. In particular, it not only takes advantage of the power of parallelism, but also avoids the problem that arises when some independent computer nodes could fail at any time (Dean and Ghemawat, 2010; Rajaraman and Ullman, 2011).

MapReduce is usually implemented in an open-source software framework, namely Apache Hadoop, for distributed storage and processing over large scale datasets on computer clusters. Apache Hadoop consists of a storage part based on the Hadoop Distributed File System (HDFS) and a processing part, i.e. MapReduce. Therefore, MapReduce can be implemented in many large scale computations, which are tolerant of hardware faults (Abaker et al., 2015). Fig. 2 shows the MapReduce computation procedure.

The MapReduce framework is based on the following procedures. Given a user-defined map function *M*, each map function turns each chunk of input data into a sequence of initial key-value pairs simultaneously in parallel on different local machines. Then, the map functions process these input data to produce a set of intermediate key-value pairs, which are collected by a master controller. Specifically, all intermediate values are grouped together, are associated with the same keys and are passed to the same machines. Next, the user-defined reduce function works on one key at a time, and combines all the values associated with that key to produce a possibly smaller set of values resulting in the final key-value pairs as the output.

The MapReduce framework has been employed to solve many big data problems. For example, Triguero et al. (2015) applied the MapReduce concept to solve a data reduction problem, to filter out unrepresentative data from a given large scale dataset. Qian et al. (2015) proposed a novel hierarchical attribute reduction algorithm that used MapReduce to deal with the high dimensionality problem. Lopez et al. (2015) introduced a fuzzy rule based classification system based on MapReduce for the class imbalance problem. In addition, Bi et al. (2015) proposed a novel distributed extreme learning machine based on MapReduce for efficient learning over large scale datasets.

Despite the success obtained using MapReduce for various big data mining applications, there has been no general agreement for setting the number of mappers, which should be domain problem dependent. In addition, according to Triguero et al. (2015), there is a trade-off between computational cost (i.e. mining efficiency) and classification accuracy (mining effectiveness).

3. Big data mining procedures

The performance obtained using the distributed and MapReduce methodologies over large scale datasets in terms of mining accuracy and efficiency is examined by comparing three big data mining procedures, namely the baseline (centralized), distributed, and MapReduce procedures.

3.1. The baseline big data mining procedure

The baseline procedure for big data mining is performed on a specific single machine and the data is centralized for mining purposes. Fig. 3 shows an example of using the support vector machine (SVM) classification technique for a classification problem based dataset. First of all, the 10-fold cross validation strategy is used to split the original dataset into 90% for a training set and 10% for a testing set. Then, the training set is used to construct the SVM classifier. Finally, the testing set is fed into the SVM for the classification result.

In this study, the baseline big data mining procedure is executed on a single PC with the Windows 7 operating system, an Intel(R) Core (TM) i7-3770 CPU @ 3.40 GHz, and 16.0 GB RAM.

3.2. The distributed big data mining procedure

Differing from the baseline procedure, the distributed big data mining procedure borrows the divide-and-conquer principle where a given dataset is divided into n subsets for n computer nodes and the SVM algorithm is implemented in each computer node. In this study, different numbers of computer nodes are compared in order to figure out the effects of computer nodes on the mining accuracy and efficiency. In particular, we set n to range from 10 to 50 with an interval of 10.

Fig. 4 shows an example of a distributed big data mining procedure using 5 computer nodes. First, the original dataset is split 90% into a training set and 10% into a testing set by the 10-fold cross validation strategy. Next, the training set is divided into five non-duplicated subsets, and each subset is used to train the SVM classifier, which results in five SVM classifiers constructed individually. Then, the testing set is fed into the five SVM classifiers simultaneously. For each test sample, the five outputs produced by the five SVM classifiers respectively are combined by the majority voting combination method for the final classification result. In addition to classification accuracy, the training and classification times are measured.

It should be noted that as discussed in Section 2.1, we use the same computing platform as the baseline to accomplish the distributed data mining procedure. That is, since the SVM classifier i trained by training set i ($i = 1$–5) is executed on a single computer, the five different SVM classifiers can be constructed separately using the same computer. Then, the testing stage and results can be combined on one single machine.

3.3. The MapReduce based big data mining procedure

Fig. 5 shows an example of using five computer nodes to perform big data mining based on the MapReduce framework. In this study, a computer server is used to simulate the cloud computing environment needed to accomplish this procedure. Specifically, the

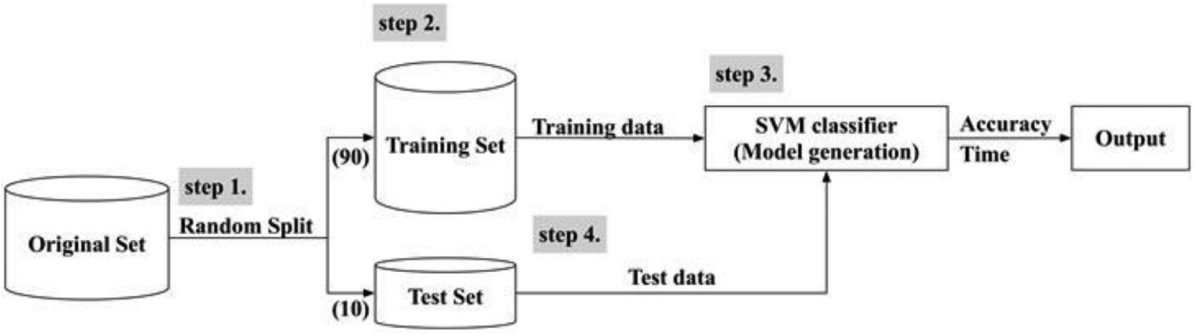

Fig. 3. Baseline big data mining procedure.

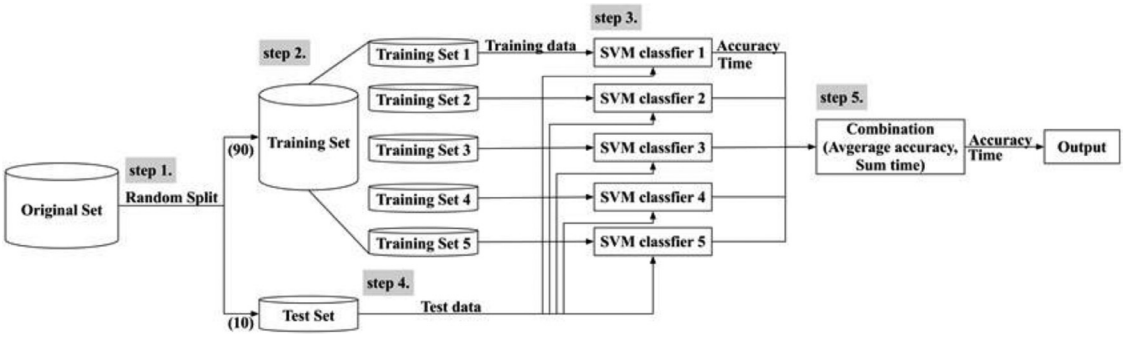

Fig. 4. Distributed big data mining procedure.

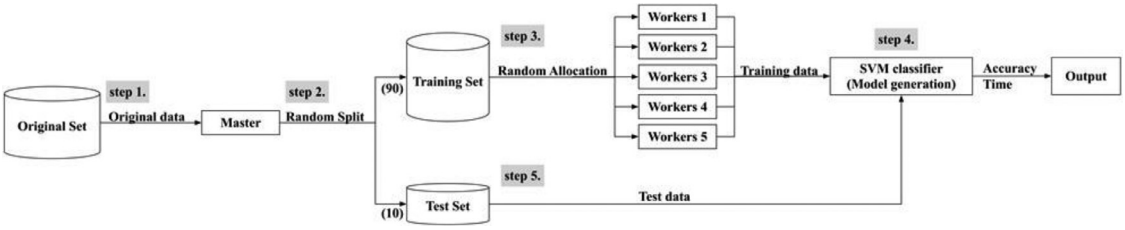

Fig. 5. MapReduce based big data mining procedure.

Table 1
Hardware and software information.

Manufacturer	Dell Inc.
Model	PowerEdge T610
Processor	Intel(R) Xeon(R) CPU E5620 @ 2.40 GHz
Host CPU	8 CPUs
Memory	196.0 GB
Host operating system	VMkernel
Virtualisation software	VMware vSphere Hypervisor (ESXi)
Number of virtual machines	1∼51 (one for the master and the others for the workers)
Guest operating system	CentOS 6.5
Guest CPU	Single-Core
Guest memory	3.0–4.0 GB
MapReduce environment	Hadoop 2.2.0
Data mining environment	Spark 0.8.1

chosen large scale dataset is located in the Hadoop HDFS based on a master and n virtual machines (i.e. workers or computer nodes) allocated for the data processing and analysis task, where n is set at 10, 20, 30, 40, and 50 for comparison. Table 1 lists the hardware and software information for the computing environment.

Unlike the distributed big data mining procedure, each of the five workers may deal with different portions of the 90% training set, which is managed by the master automatically. Therefore, the five workers have different computational complexities during the mining task. In other words, the distributed based approach focuses on partitioning the dataset per se, whereas the MapReduce based approach is used for managing the number of workers.

4. Experiments

4.1. Experimental setup

To compare the performance of the three different big data mining procedures, four large scale datasets that cover different domain problems are used. They are the KDD Cup[2] 2004 (protein homology prediction) and 2008 (breast cancer prediction), covertype[3] and person activity[4] datasets. Table 2 lists the basic information for these four datasets. The former two datasets (i.e. KDD Cup 2004 and 2008) belong to 2-class classification problems and the latter two (i.e. covertype and person activity) are multi-class classification problems.

In addition, each dataset is divided into 90% training and 10% testing sets based on the 10-fold cross validation strategy (Kohavi, 1995), for training and testing the SVM classifier, respectively. The classification accuracy of the SVM and the times for training and

[2] http://www.sigkdd.org/kddcup/
[3] http://archive.ics.uci.edu/ml/datasets/Covertype
[4] http://archive.ics.uci.edu/ml/datasets/Localization+Data+for+Person+Activity

Table 2
Basic information for the four datasets.

Datasets	No. of features	No. of samples	No. of classes
Breast cancer	117	102,294	2
Protein homology	74	145,751	2
Covertype	54	581,012	7
Person activity	8	164,860	11

Table 3
Environmental settings for the three big data mining procedures.

No. of nodes\environment	Physical environment	Virtual environment
1	Baseline procedure	MapReduce based procedure
10	Distributed procedure	
20		
30		
40		
50		

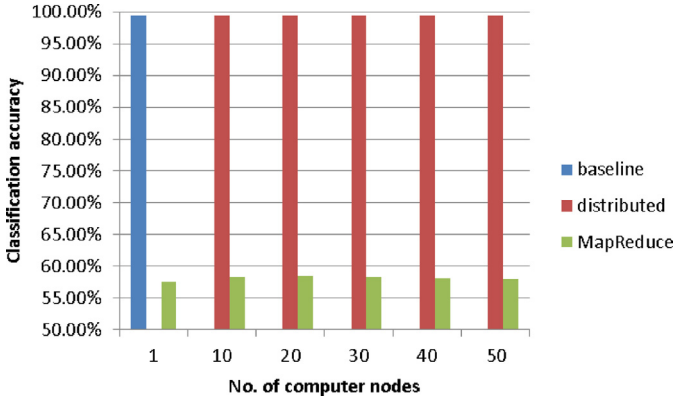

Fig. 6. Classification accuracy of three big data mining procedures over the breast cancer dataset.

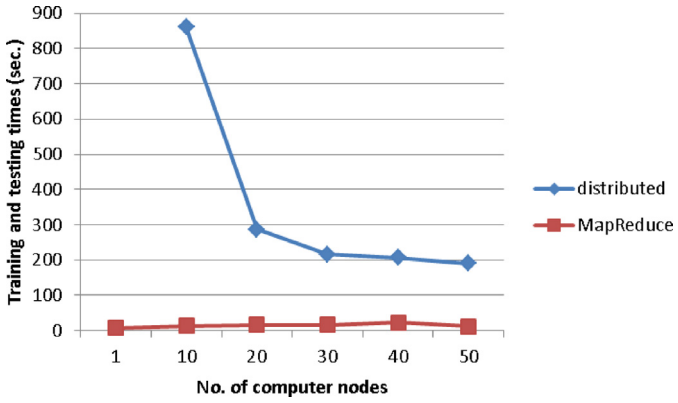

Fig. 7. Classifier training and testing times of the distributed and MapReduce based procedures over the breast cancer dataset.

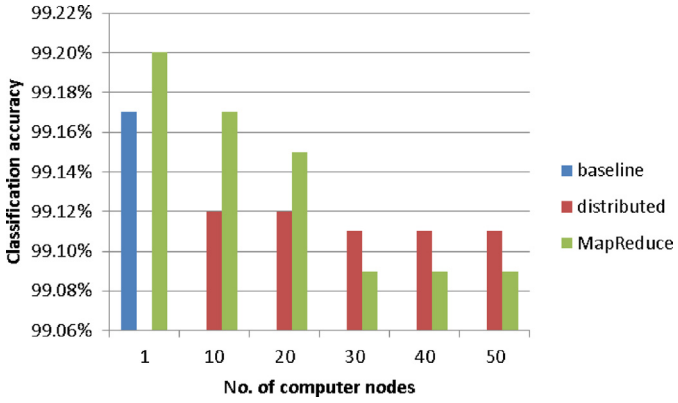

Fig. 8. Classification accuracy of three big data mining procedures over the protein homology dataset.

testing the SVM of the three different procedures for evaluation metrics are examined. Note that the computing environments of the three procedures are described in Section 3.

Table 3 shows the environmental settings of the three procedures for comparison. For the baseline procedure, only one PC is used for the big data mining task. The distributed procedure is based on dividing the training set into 10, 20, 30, 40, and 50 subsets, where one computer node is associated with one specific subset for the classifier training task. On the other hand, the MapReduce based procedure is implemented by a computer sever (c.f. Table 1) with the settings of 1, 10, 20, 30, 40, and 50 virtual machines (i.e. computer nodes) to train the classifier, respectively.

4.2. Experimental results on two-class classification datasets

4.2.1. Results on the breast cancer dataset

Fig. 6 shows the classification accuracy of the three big data mining procedures over the breast cancer dataset. We can see that the SVM classifier based on the baseline and distributed procedures can provide better performance than the MapReduce process does. In particular, the SVM obtained using the baseline and distributed procedures (based on 10–50 computer nodes) all produce 99.39% accuracy; whereas SVM obtained by the MapReduce based procedure only produces around 58% accuracy.

One reason for the poorer performance of the MapReduce based procedure may be because the breast cancer dataset is a class imbalance dataset with high dimensionalities, in which the dataset is composed of 99.4% and 0.6% data for the benign and malignant classes, respectively. According to the MapReduce framework, the SVM classifier cannot be trained effectively to efficiently distinguish between these two classes.

In contrast, Fig. 7 shows the computational costs of training and testing the SVM by the distributed and MapReduce based procedures. Note that the baseline procedure is not compared in this figure because it takes 10,223 s (i.e. nearly 3 h) to accomplish this task.

Regarding Fig. 7, we can observe that the computational costs decrease when the number of computer nodes increases based on the distributed procedures, but there is no significant reduction in the processing times using 30–50 computer nodes. Specifically, only about 3 min are required when using 50 computer nodes to train and test the SVM and this can produce the highest rate of classification accuracy (i.e. 99.39%).

For the MapReduce based procedure, the shortest time, around 7 s, comes from a single node. However, increasing the number of computer nodes does not decrease the computational cost.

4.2.2. Results on the protein homology dataset

Fig. 8 shows the classification accuracy of the three big data mining procedures for the protein homology dataset. The differences in classification performance between these three procedures are less than 0.12%. When the number of computer nodes increases from 10 to 30, there is a slight degradation in the classification accuracy of the distributed and MapReduce based procedures. However, both procedures produce similar classification accuracy when using 30–50 computer nodes.

Fig. 9 shows the SVM training and testing times for the distributed and MapReduce based procedures. The MapReduce based procedure requires the least computational cost, in which using

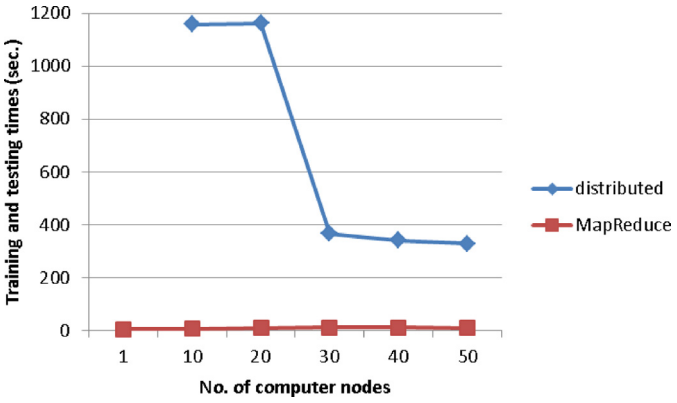

Fig. 9. Classifier training and testing times of the distributed and MapReduce based procedures over the protein homology dataset.

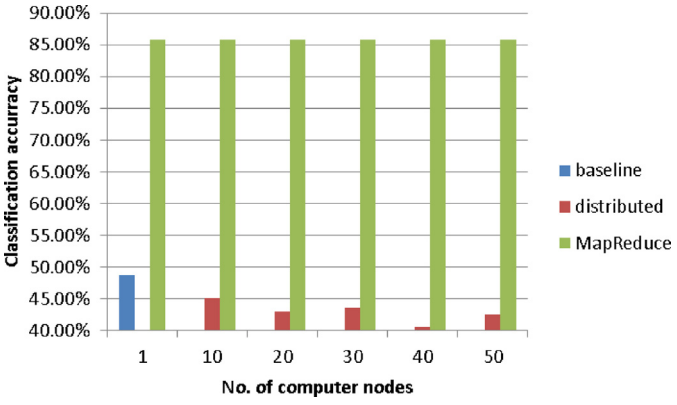

Fig. 10. Classification accuracy of three big data mining procedures over the covertype dataset.

one visual machine only takes around 6 s to accomplish the task. With the distributed procedure, similar to the previous results for the breast cancer dataset, using more computer nodes can reduce the process times.

In short, the MapReduce based procedure performs the best in terms of computational cost. However, it could be sensitive to class imbalance datasets. One possible solution for this problem is to consider some re-sampling techniques to balance the training dataset before the classifier training stage (Galar et al., 2012). On the other hand, for the distributed procedure, the processing times can be reduced if more computer nodes are used. However, the classification accuracy does not suffer significant degradation by using a large number of computer nodes. Therefore, if one considers both classification accuracy and computational cost, the distributed procedure based on about 30 computer nodes can be regarded as an optimal solution for two-class datasets.

Moreover, for the memory usage during the classifier training stage, on average the baseline, distributed (30 nodes), and MapReduce (30 nodes) procedures require 12.7 GB of RAM, 1.2 GB of RAM, and 0.9 GB of RAM, respectively. This shows that the MapReduce procedure requires the smallest memory consumption. For the classifier testing stage, the memory usages of these three procedures are not significantly different, which range from 0.3 to 0.5 GB of RAM.

4.3. Experimental results on multi-class classification datasets

4.3.1. Results on the covertype dataset

Fig. 10 shows the classification accuracy of the three big data mining procedures for the covertype dataset. The SVM classifier

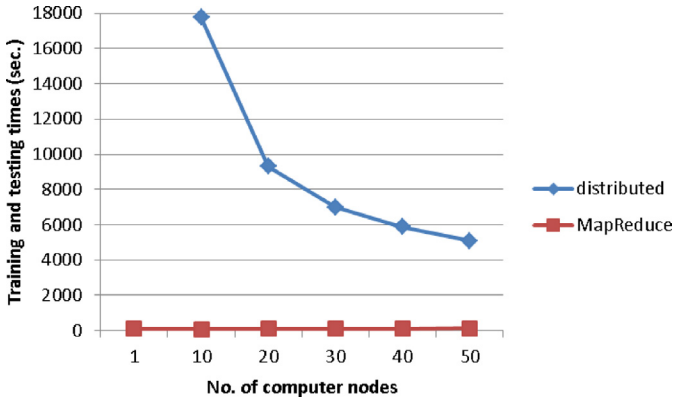

Fig. 11. Classifier training and testing times of the distributed and MapReduce based procedures over the covertype dataset.

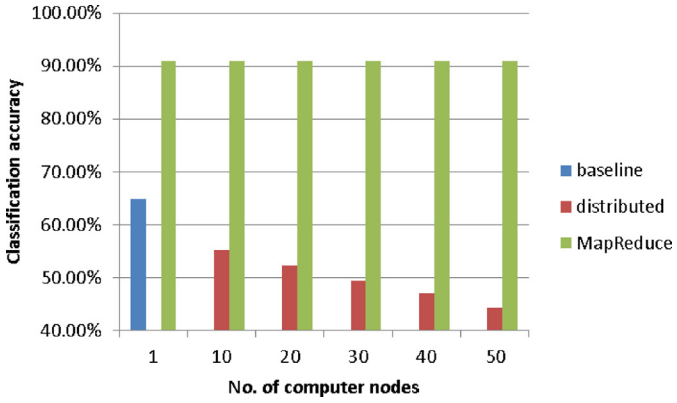

Fig. 12. Classification accuracy of three big data mining procedures over the person activity dataset.

based on the MapReduce based procedure outperforms the baseline and distributed procedure, which is different from the previous results. In addition, the classification accuracy is the same, no matter how many computer nodes are used (i.e. 85.71%). On the other hand, the SVM based on the distributed procedure demonstrates unstable performance when using different numbers of computer nodes.

Fig. 11 shows the computational costs of training and testing the SVM for the distributed and MapReduce based procedures. In this dataset, the baseline procedure takes 173,911 s (i.e. around 48 h) while about 1–5 h are required for the distributed procedure during the classifier training and testing steps. However, the MapReduce based procedure requires the least amount of training and testing time, especially when 10 computer nodes are used, requiring only 76 s.

Although the computational costs obtained using different numbers of computer nodes based on the MapReduce based procedure are similar (i.e. about 1–2 min), the processing times gradually increase when the number of computer nodes increases from 10 to 50.

These results indicate that increasing the number of computer nodes does not necessarily mean that the processing time can be reduced. This is because in the MapReduce framework using larger numbers of computer nodes creates a need to allocate the training set to more workers (i.e. computer nodes) during the computation. Therefore, more communication between different works are needed.

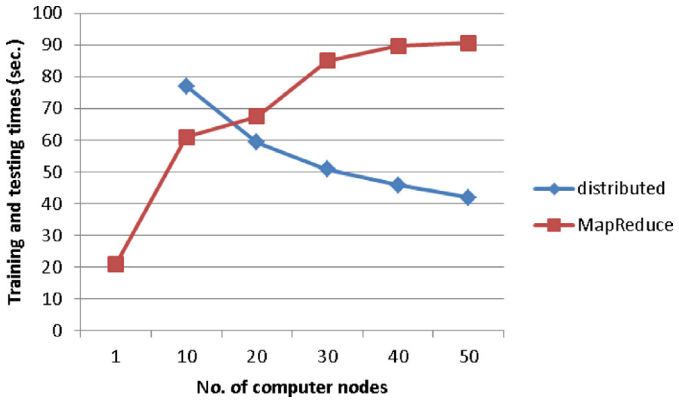

Fig. 13. Classifier training and testing times of the distributed and MapReduce based procedures over the person activity dataset.

4.3.2. Results for the person activity dataset

Fig. 12 shows the classification accuracy of the three big data mining procedures over the person activity dataset. We can see that when using the distributed procedure the classification accuracy gradually decreases as the number of computer nodes increases. In particular, the degradation in performance is more obvious than with the covertype dataset when more computer nodes are used.

In contrast, the MapReduce based procedure allows the SVM classifier to produce the highest rate of classification accuracy and the classification accuracy remains stable no matter how many computer nodes are used, at 90.91%. As this dataset is an 11-class classification domain problem, which can be regarded as a complex dataset like the covertype dataset, these results demonstrate the suitability of using the MapReduce based procedure for this type of large dataset.

Fig. 13 shows the computational costs of the distributed and MapReduce based procedures. The baseline procedure takes 542 seconds to accomplish this task. In the distributed procedure, as the number of computer nodes increases, the computational cost is reduced, but it becomes larger in the MapReduced based procedure. This indicates that there is no need to use a large number of computer nodes in this dataset to ensure classification accuracy and processing times. Specifically, one single machine can be used in the MapReduce based procedure to make the SVM produce the highest accuracy rate and require the least processing time, i.e. 21 s.

The covertype and person activity datasets contain very large numbers of data samples and they are multi-class classification domain problems, which are much larger and more complex than two-class datasets used in Section 4.2. For this type of big dataset, the MapReduce based procedure (by one to ten visual machines) is the best choice since it can allow the classifier to provide the highest rate of classification accuracy and requires the least amount of processing time compared with the baseline and distributed procedures. In other words, the MapReduce based procedure can deal with more complex and larger volumes of data more effectively and efficiently than the conventional baseline and distributed procedures. This indicates that the MapReduce based procedure is a better solution for big data mining, especially when the datasets contain some highly complex characteristics, such as a very large volume of data samples and multi-class classification problems.

For the memory consumption during the classifier training stage, on average the baseline, distributed (20 nodes), and MapReduce (20 nodes) procedures require 15.7 GB of RAM, 1.5 GB of RAM, and 1.1 GB of RAM, respectively. On the other hand, for the classifier testing stage, which is similar to the results of Section 4.2, the memory usages of these three procedures require about 0.3–0.5 GB of RAM.

4.4. Further comparisons

Data pre-processing is an important step in the knowledge discovery in databases (KDD) process (Pyle, 1999). Thus we further examine how performing this data pre-processing step can affect the performances of these three big data mining procedures. Here, instance selection (García et al., 2012) is considered in the data pre-processing step. In particular, the aim of instance selection is to filter out some noisy data from a given training set leading to a reduced training set, which is composed of more representative training data, to make the classifier training stage more effective and efficient. In this experimental study, two instance selection algorithms are employed for comparison, DROP3 (Wilson and Martinez, 2000) and the genetic algorithm (GA) (Cano et al., 2003).

Since the current version of Spark does not provide a built-in instance selection module, in order to fairly compare the performances of these three procedures for pre-processed big data, each dataset is first divided into 90% training and 10% testing sets by 10-fold cross validation. Then, instance selection is performed over the training dataset in the baseline computing environment. Next, the reduced training dataset is used to replace the original training set in the baseline, distributed, and MapReduce based procedures, respectively (c.f. Figs. 3–5). The classification results obtained combining DROP3 and GA with the three procedures over the four chosen datasets are shown in Figs. 14–17, respectively.

These results are consistent with the previous ones obtained with the MapReduce based procedure does not perform well in the class imbalance dataset (i.e. the breast cancer dataset) despite a number of noisy data being removed. Using the other datasets, the MapReduce based procedure performs significantly better than the baseline and distributed procedures and it can provide the same or very similar performance no matter how many computer nodes are used. Specifically, the differences in performance when using 1–50 computer nodes obtained via the MapReduce based procedure are less than 0.2% for DROP3 and 0.5% for GA over the protein homology dataset.

On the other hand, using different instance selection algorithms is likely to affect the performance of the big data mining procedures. For example, the distributed and MapRedcue based procedures combined with GA outperform the ones combined with DROP3 for the person activity and breast cancer datasets, respectively. However, the MapReduce based procedure combined with DROP3 performs slightly better than the one combined with GA for the protein homology dataset.

5. Conclusion

Big data mining can be tackled efficiently under a parallel computing environment. In general, two different methodologies can be employed. The first one is based on the distributed procedure, which focuses on the data parallelism principle to manually divide a given large scale dataset into a number of subsets, each of which is handled by one specific learning model implemented on one single machine. The final result is obtained by combining the outputs generated by the learning models. The second is the MapReduce based procedure, which is based on the map and reduce process where the number of maps can be user-defined, but are all controlled by a master to automatically manage the utility and consumption of computing resources for a computer cluster. Then, the reduce function combines the outputs of the maps for the final result.

The aim of this paper is to examine the mining performance and efficiency of the distributed and MapReduce based procedures

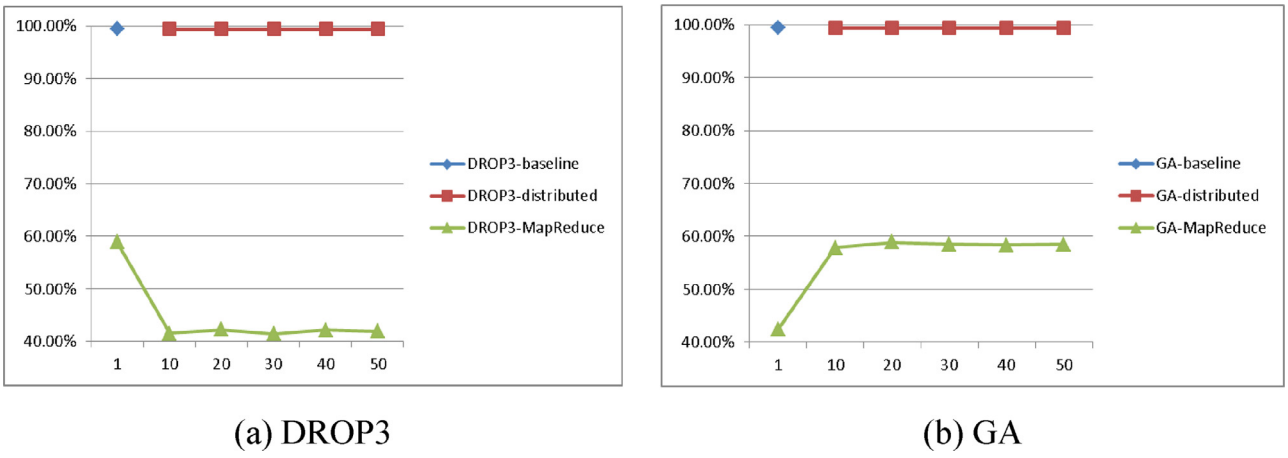

Fig. 14. Classification performances obtained combining DROP3/GA with the three procedures over the breast cancer dataset.

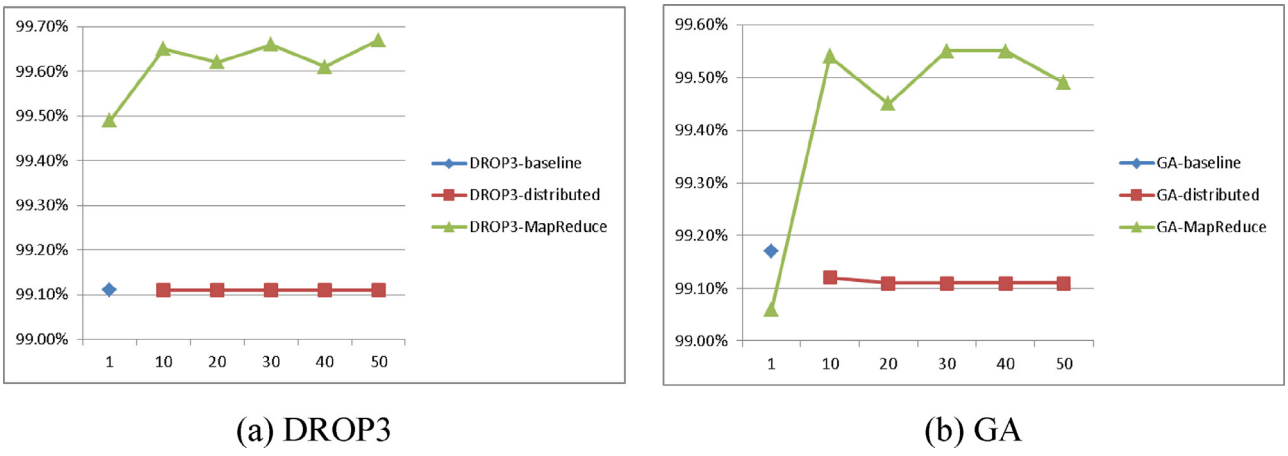

Fig. 15. Classification performances obtained combining DROP3/GA with the three procedures over the protein homology dataset.

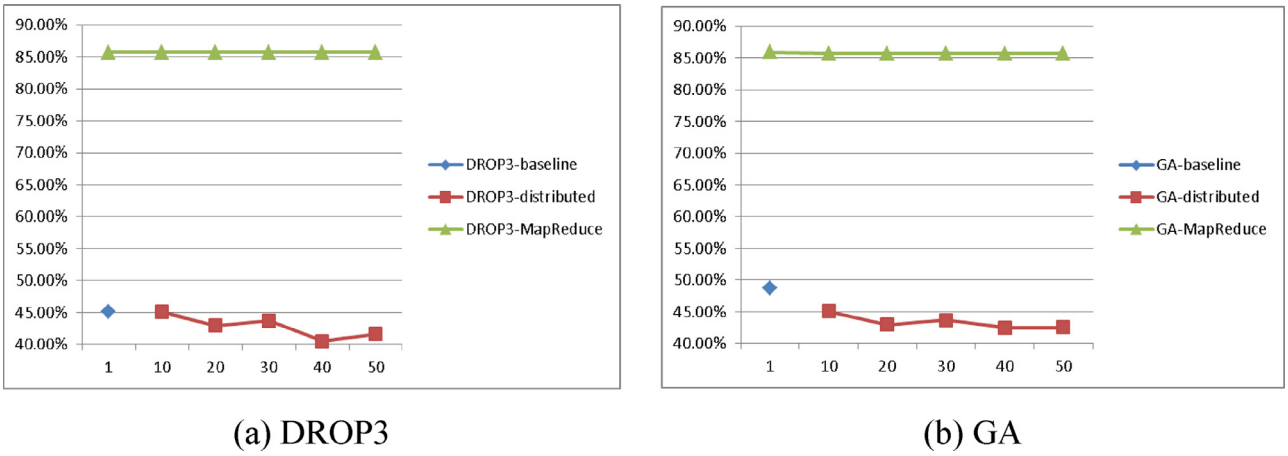

Fig. 16. Classification performances obtained combining DROP3/GA with the three procedures over the covertype dataset.

over big data problems. Our experimental results based on four large scale datasets show that the MapReduce based procedure performs very stably in terms of mining accuracy no matter how many computer nodes are used and it can allow the SVM classifier to provide the highest rate of classification accuracy with the exception of the class imbalance dataset. In addition, the least amount of processing time is required for training and testing the SVM, although increasing the number of computer nodes may slightly increase the processing times. It is found that using one to ten computer nodes is the better choice.

For the distributed procedure, when the number of computer nodes increases, the classification accuracy gradually decreases. However, the processing time shows the opposite result. In other words, it displays more partitions from the original dataset, meaning that each partition becomes smaller, and thus the processing time for each computer node is reduced. On the other hand, a

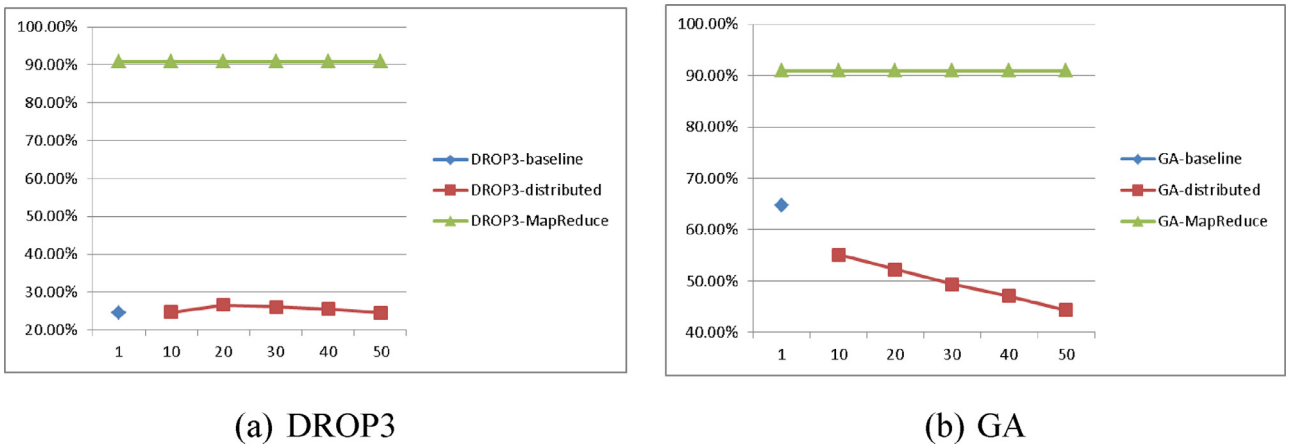

Fig. 17. Classification performances obtained combining DROP3/GA with the three procedures over the person activity dataset.

learning algorithm that only uses a small portion of the training data can make it perform worse. In short, in the distributed procedure there is a trade-off between the processing time and classification accuracy, but this is not obvious in the MapReduce based procedure.

Several issues can be considered in future work. First, more large scale datasets containing various amount of data samples, different numbers of features (i.e. dimensionalities), and different feature types including categorical, numerical, and mixed data types can be used for further comparisons. Second, in addition to constructing the SVM classifier, the performances of using other classification techniques under the three different procedures can be examined. Last but not least, it would be useful to investigate the effect of using different computing hardware environments on the three different procedures.

References

Abaker, I., Hashem, T., Yaqoob, I., Anuar, N.B., Mokhtar, S., Gani, A., Khan, S.U., 2015. The rise of "big data" on cloud computing: review and open research issues. Inf. Syst. 47, 98–115.

Bi, X., Zhao, X., Wang, G., Zhang, P., Wang, C., 2015. Distributed extreme learning machine with kernels based on MapReduce. Neurocomputing 149, 456–463.

Cano, J.R., Herrera, F., Lozano, M., 2003. Using evolutionary algorithms as instance selection for data reduction: an experimental study. IEEE Trans. Evol. Comput. 7 (6), 561–575.

Coulouris, G., Dollimore, J., Kindberg, T., Blair, G., 2011. Distributed Systems: Concepts And Design, 5th ed. Addison-Wesley.

Dean, J., Ghemawat, S., 2010. Map reduce: a flexible data processing tool. Commun. ACM 53 (1), 72–77.

Fan, W., Bifet, A., 2012. Mining big data: current status, and forecast to the future. ACM SIGKDD Explor. Newslett. 14 (2), 1–5.

Fernandez, A., del Rio, S., Lopez, V., Bawakid, A., del Jesus, M.J., Benitez, J.M., Herrera, F., 2014. Big data with cloud computing: an insight on the computing environment, MapReduce, and programming frameworks. Wiley Interdiscip. Rev. 4 (5), 380–409.

Galar, M., Fernandez, A., Barrenechea, E., Bustince, H., Herrera, F., 2012. A review on ensembles for the class imbalance problem: bagging-, boosting-, and hybrid-based approaches. IEEE Trans. Syst. Man Cybern. Part C 42 (4), 463–484.

García, S., Derrac, J., Cano, J.R., Herrera, F., 2012. Prototype selection for nearest neighbor classification: taxonomy and empirical study. IEEE Trans. Pattern Anal. Mach. Intell. 34 (3), 417–435.

Gottlieb, A., Almasi, G., 1989. Highly Parallel Computing. Benjamin-Cummings Publishing.

Kittler, J., Hatef, M., Duin, R.P.W., Matas, J., 1998. On combining classifiers. IEEE Trans. Pattern Anal. Mach. Intell. 20 (3), 226–239.

Kohavi, R., 1995. A study of cross-validation and bootstrap for accuracy estimation and model selection. In: International Joint Conference on Artificial Intelligence, pp. 1137–1143.

Lopez, V., del Rio, S., Benitez, J.M., Herrera, F., 2015. Cost-sensitive linguistic fuzzy rule based classification systems under the MapReduce framework for imbalanced big data. Fuzzy Sets Syst. 258, 5–38.

Mayer-Schonberger, V., Cukier, K., 2014. Big Data: A Revolution That Will Transform How We Live, Work, And Think. Eamon Dolan/Mariner Books.

Park, B., Kargupta, H., 2002. Distributed data mining: algorithms, systems, and applications. In: Ye, N. (Ed.), Data Mining Handbook. Oxford University Press, pp. 341–358.

Peteiro-Barral, D., Guijarro-Berdinas, B., 2013. A survey of methods for distributed machine learning. Prog. Artif. Intell. 2, 1–11.

Pyle, D., 1999. Data Preparation For Data Mining. Morgan Kaufmann.

Qian, J., Lv, P., Yue, X., Liu, C., Jing, Z., 2015. Hierarchical attribute reduction algorithms for big data using MapReduce. Knowl. Based Syst. 73, 18–31.

Rajaraman, A., Ullman, J.D., 2011. Mining Of Massive Datasets. Cambridge University Press.

Triguero, I., Peralta, D., Bacardit, J., Garcia, S., Herrera, F., 2015. MRPR: a MapReduce solution for prototype reduction in big data classification. Neurocomputing 150, 331–345.

Wilson, D.R., Martinez, T.R., 2000. Reduction techniques for instance-based learning algorithms. Mach. Learn. 38, 257–286.

Wu, X., Zhu, X., Wu, G.-Q., Ding, W., 2014. Data mining with big data. IEEE Trans. Knowl. Data Eng. 26 (1), 97–107.

Zaki, M.J., 2000. Parallel and distributed data mining: an introduction. Lect. Notes Comput. Sci. 1759, 1–23.

Zheng, L., Li, L., Duan, L., Lu, K., Shi, Z., Wang, M., Wu, W., Luo, P., 2012. Distributed data mining: a survey. Inf. Technol. Manage. 13, 403–409.

Zhou, Z.-H., Chawla, N.W., Jin, Y., Williams, G.J., 2014. Big data opportunities and challenges: discussions from data analytics perspectives. IEEE Comput. Intell. Mag. 9 (4), 62–74.

Dr. Chih-Fong Tsai received a PhD at School of Computing and Technology from the University of Sunderland, UK in 2005. He is now a professor at the Department of Information Management, National Central University, Taiwan. He has published more than 50 technical publications in journals, book chapters, and international conference proceedings. He the Highly Commended Award (Emerald Literati Network 2008 Awards for Excellence) from *Online Information Review* ("A Review of Image Retrieval Methods for Digital Cultural Heritage Resources"), and the award for top 10 cited articles in 2008 from *Expert Systems with Applications* ("Using Neural Network Ensembles for Bankruptcy Prediction and Credit Scoring"). His current research focuses on multimedia information retrieval and data mining.

Dr. Wei-Chao Lin is an associate professor at the Department of Computer Science and Information Engineering, Asia University, Taiwan. His research interests are machine learning and artificial intelligence applications.

Dr. Shih-Wen Ke is an assistant professor at the Department of Information and Computer Engineering, Chung Yuan Christian University, Taiwan. His research covers information retrieval, machine learning, and data mining.