Data Partitioning for Incremental Data Mining

Nittaya Kerdprasop and Kittisak Kerdprasop

School of Computer Engineering, Suranaree University of Technology
111 University Avenue, Muang District, Nakorn Ratchasima 30000, THAILAND
nittaya , kerdpras @ccs.sut.ac.th

ABSTRACT

Data repositories of interest in data mining applications can be very large. Many of the existing learning algorithms do not scale up to extremely large data set. One approach to deal with this problem is to apply the concept of incremental learning. However, incremental data mining is not the same as incremental machine learning. The former handles one subset of data at a time, whereas the latter handles a single data instance at a time. The size of data subset determines both the performance and speed of the mining process. We thus focus the study on the partitioning of a data into a proper subset and propose an algorithm to return a data subset for both classification and association mining tasks[1]. We also perform a set of experiments to observe the behavior of classification and association data mining on various data partitioning. The experimental results confirm our criteria on data partitioning.

Keywords: data mining, incremental, data partitioning

1 INTRODUCTION

Data mining is the process of extracting useful information such as previously unknown patterns or association hidden in a large data set [4]. Recent advances in digital information storage and data acquisition technologies have made it possible to acquire and store large volumes of data. Therefore, data repositories of interest in data mining applications are normally very large. Many of the existing mining algorithms do not scale up to extremely large data sets. One approach to deal with this problem is to partition the huge data set into several subsets of manageable size, then learn (probably in parallel) from each subset and finally combine the learning results [3].

Another approach is to employ the incremental machine learning paradigm. Incremental machine learning is the technique to avoid retaining all training data in main memory. Instead, the learning algorithm learns from one data instance at a time and tune the result accordingly [2]. However, incremental data mining is not the same as incremental machine learning. Incremental data mining handles subsets of data one set at a time, not just a single data instance as in the incremental machine learning [16]. Thus, partitioning the data set to a proper subset (or sample) size is certainly beneficial the incremental mining to reach its high accuracy in an acceptable period of time.

We propose an algorithm to partition the original large data set into a manageable and yet learning-effective data subset. We also perform experiments on learning performance of different data partitions on the two common data mining tasks: classification and association. On data classification, we investigate learning curves of classifiers on various data partitions. For the derivation of association rules, we compare the set of rules obtained from each data partitions against the rules derived from the whole complete data set.

This paper is organized as follows. Section 2 gives an overview of incremental data mining. Section 3 describes the data-partitioning algorithm. Section 4 explains the experimental setup. Section 5 presents the results and a discussion. Section 6 concludes our work.

2 INCREMENTAL DATA MINING

Machine learning techniques can be broadly categorized as either batch or incremental [5]. Batch learning examines a whole collection of data set and induces a learning result. In incremental learning, data subsets D_1 , D_2 , … , D_n are assumed to become available to the learner at discrete time intervals, and the learner is also assumed being unable to store collectively all the data fragments. Thus, it can only maintain and update its learning result as new data fragment becomes available.

Sutton and Whitehead [11] have distinguished two kinds of incremental learning: weakly and strictly incremental method. A learning method is weakly incremental if it requires additional memory and computation in order to process one additional data instance. Examples of weakly incremental learners are ID5 [12,13,14] (an incremental version of ID3 [8]) and nearest neighbor algorithms. A learning method is strictly incremental if its memory and computation (per data instance) requirements do not increase with the number of instances. The learners in this category are STAGGER[10] and most connectionist learning methods.

[1] The work reported in the paper was supported by the grant from the National Electronics and Computer Technology Center (NECTEC).

Recent research on learning ensembles of classifiers [3] is relevant to incremental data mining. Learning ensembles of classifiers, such as bagging [1] and boosting, generates multiple versions of classifiers by running the learning algorithm many times on a set of re-sampled data. The classification results are combined using a majority vote. Each version of the classifier is generated from a sample of the original data set, and each data instance can be used in many samples. To adopt the ensemble method to the setting of incremental mining, -- for instance, Learn++ algorithm [7] -- each data instance in the original data set is partitioned into only one subset and used only once in the learning process. As mentioned in the first section that incremental data mining learns a subset – not just a single data instance, we are however still of limited knowledge about what proper size data partitioning should be. Therefore, we design an algorithm to generate a proper data subsets and test the algorithm on different data sizes to observe the efficiency of incremental data mining.

3 DATA PARTITIONING ALGORITHM

This section describes the algorithm to partition the large data set into a manageable and proper size. The algorithm returns the data partition for both the classification task and association task.

Algorithm Data_Partition

Input: (1) A relational database R.
(2) Predictive level l, the default predictive level can be raise to a 'high' level.
(3) Maximum number of instances, m, that data mining tool can handle.

Output: D_C = a data subset for classification, and
D_A = a data subset for association

Steps:

1. $\mu_{default} = 0.1$, $\mu_{high} = 0.3$
   /* Set parameter for the classification data subset */
2. $\gamma_{default} = 0.2$, $\gamma_{high} = 0.5$
   /* Set parameter for the association data subset */
3. *Sampling_size_classification* = min{ m , ($\mu_{default}$ * number of instances in R) }
   /* Sampling size for classification task at the default predictive level */
4. *Sampling_size_association* = min{ m , ($\gamma_{default}$ * number of instances in R) }
   /* Sampling size for association task at the default predictive level */
5. If l = 'high' then
   5.1 *Sampling_size_classification* = min{ m , (μ_{high} * number of instances in R) }
   5.2 *Sampling_size_association* = min{ m , (γ_{high} * number of instances in R) }
6. return
   D_C = { r_i | r_i = *Sampling(R)* , i = 1,2, …, *Sampling_size_classification* }
   D_A = { r_i | r_i = *Sampling(R)* , i = 1,2, …, *Sampling_size_association* }
   /* *Sampling(R)* is the random sampling without replacement from the database R */

□

4 EXPERIMENTS

We design the experiments to study the effect of varying the data partitioning on the efficiency of data mining. The two kinds of data mining task being explored are classification and association. On classification, the algorithms J48 and naïve Bays are selected as a benchmark to test the quality of each data partition. J48 is a re-implement [15] of C4.5 [9], which is the most well-known decision tree-based classification algorithm. The advantage of tree-based classifier is its simple and comprehensible representation format. Naïve Bayes is a statistical classifier that can learn the concept rapidly. This high learning rate property makes naïve Bayes a candidate algorithm for incremental data mining. Therefore, we decide to test the classification accuracy on these two classification algorithms.

The accuracy is estimated on the basis of number of test instances correctly classified by the induced classifier. The estimation method that we use is the holdout method in which 66% of the data instances is used for the training purpose and the remaining 34% is used as the test set.

For the task of association rule derivation, we employ the APRIORI algorithm [15]. The criteria to test the rule-deriving efficiency on each data partition is the number of association rules that match the rules derived from the whole data set. The two data sets -- mushroom and connect-4-game – used in our experiments are taken from the UCI repository [6].

5 RESULTS AND DISCUSSION

Table 1 and 2 show the learning results on classification and association rule derivation, respectively. The learning curves of J48 and naïve Bayes on each data set are illustrated in Figure 1. Figure 2 graphically compares the quality of association rule derivation on various data partitions.

The classification results reveal the fast-learning characteristic of naïve Bayes algorithm. It requires only 5-10 % of the data set to reach its highest learning accuracy. This fast-learning property is the major ingredient on incremental data mining. When taking association into consideration together with the classification, we may infer from the experimental results that the appropriate partitioning (or windowing) should be at the 10% of the whole data set. These results agree with our heuristic of setting the threshold parameter in the range 0.1-0.5 for the expected proper size of sample. In the distributed setting in which the exact size of the data set could not be guessed in advance, the fixed amount of 800-1,000 instances should give the satisfiable learning result.

Table 1: The classification accuracy on different data partitioning

Data partitioning	Accuracy tested on mushroom data		Accuracy tested on connect-4-game data	
	naïve Bayes (% correct classification)	J48 (% correct classification)	naïve Bayes (% correct classification)	J48 (% correct classification)
1 %	53.5714 %	57.1429 %	60.8696 %	56.9565 %
5 %	61.1511 %	61.5705 %	68.1462 %	68.6684 %
10 %	60.6498 %	64.2599 %	71.0057 %	74.1837 %
20 %	61.1212 %	70.5244 %	71.3975 %	75.8163 %
30 %	61.3993 %	68.2750 %	70.9041 %	76.9700 %
40 %	62.6244 %	69.1403 %	72.7471 %	78.8202 %
50 %	61.7221 %	64.7612 %	71.5107 %	79.1990 %
60 %	63.0277 %	66.5862 %	72.2392 %	79.6691 %
70 %	63.8056 %	67.9938 %	72.1500 %	81.6966 %
80 %	63.0317 %	68.0090 %	72.2627 %	82.1724 %
90 %	62.3492 %	66.0901 %	72.1956 %	83.6405 %
100 %	63.9884 %	60.0796 %	72.1332 %	79.0814 %

Table 2: The quality of association-rule derivation on different partitioning

Data partitioning	mushroom dataset [1]		connect-4-game dataset [2]	
	number of instances	number of rules matched [3]	number of instances	number of rules matched [3]
1 %	81	27	675	53
5 %	406	27	3,377	90
10 %	812	62	6,755	88
20 %	1,624	62	13,511	92
30 %	2,437	62	20,267	95
40 %	3,249	65	27,022	97
50 %	4,062	66	33,778	98
60 %	4,874	100	40,534	98
70 %	5,686	79	47,289	99
80 %	6,499	100	54,045	99
90 %	7,311	100	60,801	99

[1] The complete mushroom data set contains 8,124 instances.
[2] The complete connect-4-game contains 67,557 instances.
[3] Number of rules that matched with the association rules derived from the complete data set.

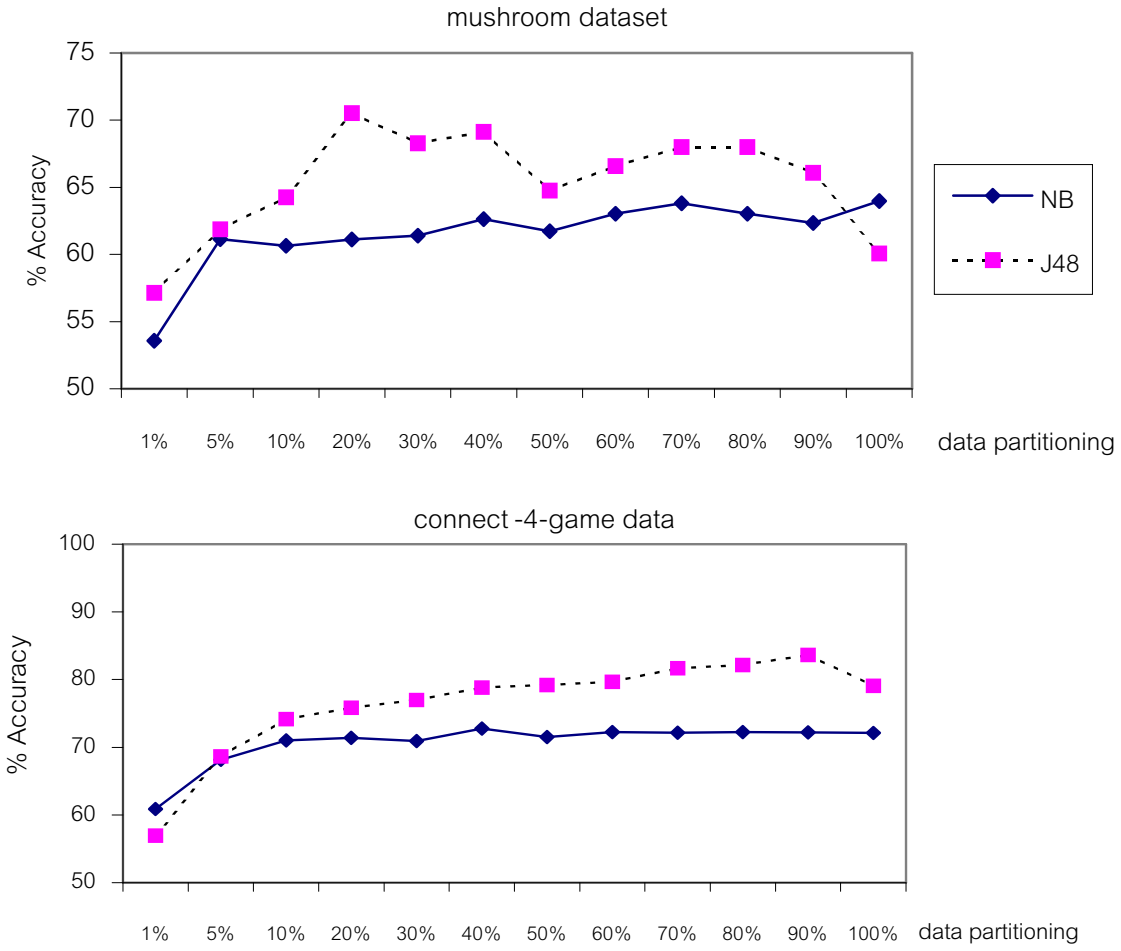

Figure 1: Learning curves on classifying the incremental mushroom data set and connect-4-game data set

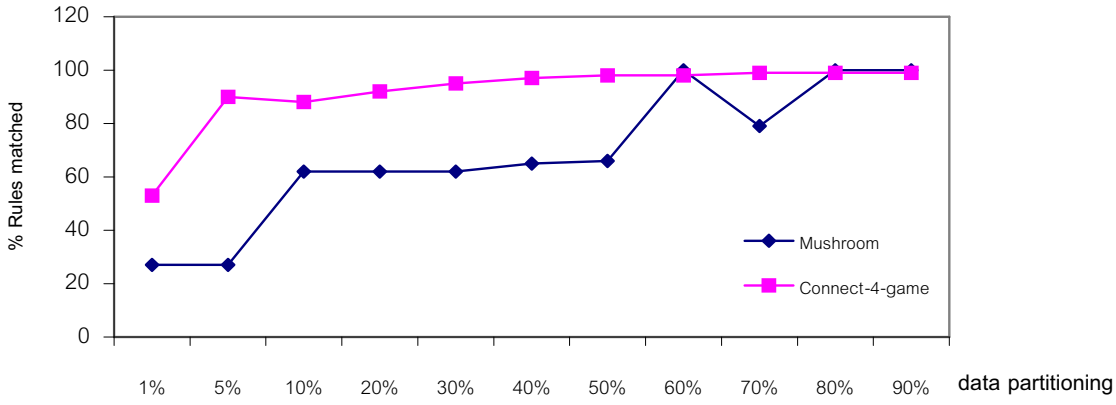

Figure 2: The comparison on quality of association rules derived from each data partitioning

6 CONCLUSION

Recent advances in data storage and acquisition techniques have made it possible to produce increasingly large data repositories. Many of the existing mining algorithms do not scale up to extremely large data sets. One approach to this problem is to partition the data set into several subsets of manageable size, then learn in parallel or incrementally from each subset. The partitioning of data set into appropriate size is the main focus of our study. We propose an algorithm to use the heuristic to do the sampling for the proper size of data subsets. We perform experiments on the two data mining tasks -- classification and association -- using mushroom and connect-4-game data sets. We can conclude from the results that partitioning the data set at the threshold level 0.1-0.5 yields an acceptable classification accuracy, and moderate to high quality association rules.

REFERENCES

[1] L. Breiman, "Arcing classifiers", Annals of Statistics, 26, 1998.

[2] S.H. Clearwater, T.P. Cheng, H. Hirsh, and B.G. Buchanan, "Incremental batch learning", Proceedings of the Sixth International Workshop on Machine Learning, Morgan Kaufmann, 1989.

[3] T.G. Dietterich, "Machine learning research: Four current directions", AI Magazine, 18(4), 1997, 97-136.

[4] U.M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy, editors, *Advances in Knowledge Discovery and Data mining*, AAAI/MIT Press, Cambridge, MA, 1996.

[5] M. Harries, C. Sammut, and K. Horn, "Extracting hidden context", Machine Learning, 36(2), 1998, 101-126.

[6] C.J. Merz and P.M. Murphy, Uci Repository of machine learning databases, 1996. [http://www.ics.uci.edu/ ~mlearn/MLRepository.html]

[7] R. Polikar, L. Udpa, S. Udpa, and V. Honavar, "Learn++: An incremental learning algorithm for multilayer percentron networks", Proceedings of the IEEE Conference on Acoustic, Speech, and Signal Processing (ICASSP), 2000.

[8] J.R. Quinlan, "Induction of decision trees", Machine Learning, 1, 1986, 81-106.

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

[10] J.C. Schlimmer and R.H. Granger, "Incremental learning from noisy data", Machine Learning, 1, 1986, 317-354.

[11] R.S. Sutton and S.D. Whitehead, "Online learning with random representations", Proceedings of the Tenth International Conference on Machine Learning, Morgan Kaufmann, 1993, 314-321.

[12] P. Utgoff, "ID5: An incremental ID3", Proceedings of the Fifth International Conference on Machine Learning, Morgan Kaufmann, 1988, 107-120.

[13] P. Utgoff, "Incremental induction of decision trees", Machine Learning, 4, 1989, 161-186.

[14] P. Utgoff, "An improved algorithm for incremental induction of decision trees", Proceedings of the Eleventh International Conference on Machine Learning, Morgan Kaufmann, 1994, 318-325.

[15] I.H. Witten and E. Frank, *Data Mining: Practical Machine Learning Tools and Techniques with java Implementations,* Morgan Kaufmann, San Francisco, 2000. [software accessible via the URL http://www.cs.waikato. ac.nz/ml/weka]

[16] X. Wu and W. Lo, "Multi-layer incremental induction", Proceedings of the Fifth Pacific Rim International Conference on Artificial Intelligence, Springer-Verlag, 1998.