Bayesian Network Structure Learning from Big Data: A Reservoir Sampling Based Ensemble Method

Yan Tang, Zhuoming Xu(✉), and Yuanhang Zhuang

College of Computer and Information, Hohai University, Nanjing 210098, China
{tangyan,zmxu,yuanhangzhuang}@hhu.edu.cn

Abstract. Bayesian network (BN) learning from big datasets is potentially more valuable than learning from conventional small datasets as big data contain more comprehensive probability distributions and richer causal relationships. However, learning BNs from big datasets requires high computational cost and easily ends in failure, especially when the learning task is performed on a conventional computation platform. This paper addresses the issue of BN structure learning from a big dataset on a conventional computation platform, and proposes a reservoir sampling based ensemble method (RSEM). In RSEM, a greedy algorithm is used to determine an appropriate size of sub datasets to be extracted from the big dataset. A fast reservoir sampling method is then adopted to efficiently extract sub datasets in one pass. Lastly, a weighted adjacent matrix based ensemble method is employed to produce the final BN structure. Experimental results on both synthetic and real-world big datasets show that RSEM can perform BN structure learning in an accurate and efficient way.

Keywords: Bayesian network structure learning · Reservoir sampling · Ensemble method · Probabilistic approximation · Big data

1 Introduction

A Bayesian network (BN) [1] is a probabilistic directed acyclic graphical model for representing multivariate probability distributions. BNs have been widely applied to various forms of reasoning in many domains such as Health Care, Finance and Transportation [2–4]. With the increasing availability of big datasets in science, government and business, BN learning from big datasets is potentially more valuable than learning from conventional, small datasets as big data contain more comprehensive probability distributions and richer causal relationships. However, learning BNs from big datasets requires high computational cost [5], easily ending in failure. Facing this challenge, one roadmap is performing the learning task on a big data processing platform using Hadoop or Spark, such as the MapReduce based method proposed by Fang et al. [6] and our previous work [7]. But such a platform is not affordable for all institutions. Therefore, an

H. Gao et al. (Eds.): DASFAA 2016 Workshops, LNCS 9645, pp. 209–222, 2016.
DOI: 10.1007/978-3-319-32055-7_18

alternative is first sampling sub datasets from the big dataset using probabilistic approximation and then learning a BN from the sampled, small sub datasets on a conventional computation platform. This study adopts the second roadmap. Since the most important and challenging step of BN learning is finding the network structure, this paper addresses the issue of sampling-based BN structure learning from a big dataset on a conventional computation platform.

We argue that to achieve Bayesian network structure learning from big data using a conventional computation platform, a big dataset needs to be appropriately sampled into several sub datasets with much smaller sizes, and an ensemble method is necessary for effectively combining the BN structures learned from the sub datasets. Hence a reservoir sampling based ensemble method, called RSEM, is proposed in this paper. The main ideas of RSEM are as follows. We introduce a minimal sample size (MSS) for sub dataset extraction, which can keep DAG-faithfulness [8] of the sub datasets, and design a greedy algorithm for calculating MSS, aimed at achieving a trade-off between learning accuracy and computational efficiency. According to the calculated MSS, we adopt a fast reservoir sampling method based on our proposed notion data reservoir index (DRI) to efficiently extract sub datasets in one pass. Lastly, we employ an ensemble method using a BDeu score [9] based weighted adjacent matrix to combine the BN structures learned from the sub datasets and produce the final BN structure in an approximate but sufficiently accurate way.

Our proposed method has been implemented using R software environment on a conventional computation platform. To validate the effectiveness of the method, we conducted experiments on both three synthetic big datasets and one real-world big dataset. The experimental results show that RSEM can sample appropriate sub datasets from big datasets by means of the calculated MSS, and perform Bayesian network structure learning from big datasets in an accurate and efficient way.

The rest of the paper is organized as follows: Sect. 2 is related work. The proposed method including algorithms is presented in Sect. 3. After giving experimental results and discussion in Sect. 4, we conclude this work in the final section.

2 Related Works

The notion DAG-faithful is the introduced is the work of TPDA algorithm [8]. A dataset is DAG-faithful if its underlying probabilistic model is DAG structured. This condition makes a dataset suitable for BN learning. The fundamental assumption of this research is that given a sufficiently large DAG-faithful dataset, its DAG-faithful sub datasets can be used to approximate the learning on the whole dataset.

In a Bayesian network, the Markov blanket (MB) of a node includes its parents, its children and the children's parents [10]. The MB of a node contains all the variables that shield the node from the rest of the network and is the only knowledge needed to predict the behavior of the node. Many algorithms like MMHC [11] were proposed to learn BN structure. An important property

of a BN is its Average Markov blanket size, denoted as AMBS, which is defined as Eq. (1).

$$AMBS = \sum_{i=1..N} MBS_i/N \tag{1}$$

where is the Markov blanket size of node i and is the total number of nodes in the network. AMBS can measure the complexity of a BN.

Structure Hamming distance (SHD), a metric introduced by Tsamardinos et al. [11], is defined as the number of the following operators required to make the network match: add or delete an undirected edge, and add, remove, or reverse the orientation of an edge [11]. It has become a widely used metric for measuring structure difference between two networks and evaluating the quality of the learned network. Small SHD indicates high learning accuracy. The number of correctly identified edges is equal to the total number of edges in the known BN minus SHD. This paper, therefore, uses SHD to evaluate the accuracy of the learning method.

Jiang et al. [12] studied the sampling of the datasets and applied the sampled datasets to different Bayesian network classifiers to achieve better classification accuracy, which validates the effectiveness of data sampling methods for BN learning. Reservoir sampling [13] is a widely used randomized algorithm for randomly choosing samples from a big dataset which doesn't fit into main memory. We leverage reservoir sampling to efficiently sample sub datasets from a big dataset.

In machine learning, ensemble methods [14] use multiple learning methods to obtain better predictive performance than learning from any of the constituent methods. Hasna and Salma [15] proposed a weighted ensemble Bayesian network learning method for gene regulatory networks. Our previous work [16] achieved higher accuracy of BN structure learning through ensemble methods. In this paper, we continue to adopt ensemble methods for achieve better learning accuracy.

In the field of BN learning from big data, Chickering et al. [17] showed that identifying high-scoring BN from large dataset is NP-hard. Yoo et al. [18] reviewed bioinformatics and statistical methods and concluded that Bayesian networks are suitable in analyzing big datasets from clinical, genomic, and environmental domains.

Furthermore, Fang et al. [6] proposed a Map-Reduce based method for learning BN from massive datasets. Our previous work [7] adopted distributed data-parallelism techniques and scientific workflow for BN learning from big datasets to achieve better scalability and accuracy. To the best of our knowledge, most existing studies applied data parallelization techniques to whole big dataset learning, much less work had used data sampling to learn BN from big datasets for reducing learning complexity.

3 The Proposed Method

3.1 Overview of the Method

Figure 1 is the overview of our proposed reservoir sampling based ensemble method (abbreviated as RSEM) for Bayesian network structure learning from big data, which consists of the following three key steps.

Firstly, RSEM takes the big dataset as the input and uses a greedy algorithm to calculate the minimal sampling size (MSS) of extracted sub datasets for a specific learning task in the BN learning procedure.

Secondly, a fast reservoir sampling algorithm is designed to sample sub datasets with the size of MSS from the big dataset. This sampling algorithm only requires one iteration over the entire dataset.

Lastly, an ensemble algorithm (by means of a BDeu score based, weighted adjacent matrix) is adopted to merge the Bayesian networks (BNs) learned from all the sub datasets and then produce a final BN as the output.

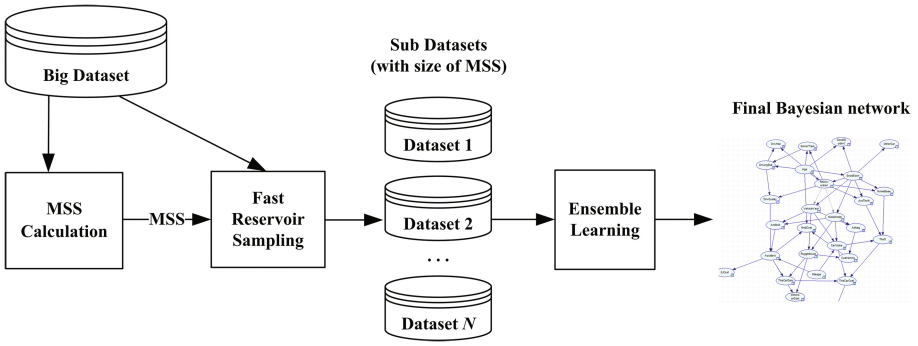


Fig. 1. Overview of the RSEM method

3.2 Calculation of MSS

Given a DAG-faithful (big) dataset with a sufficiently large size, it is reasonable to learn a BN from its sub datasets instead of the whole dataset. Learning on the sub datasets could achieve high computation efficiency and approximate the whole data learning without loss of generality. The key challenge here is the selection of sub dataset size. If the size is too small, then a poorly structured BN will be learned, otherwise, low computation efficiency as well as overfitting will occur. Thus, we introduce a novel concept called the minimal sampling size (MSS), as defined below.

Definition 1. *(minimal sampling size)Given a DAG-faithful and independent identically distributed (iid) dataset D, its minimal sampling size (MSS_D) is the*

minimal size of sub dataset that maintains DAG-faithful. MSS_D *is defined in Eq. (2):*

$$MSS_D = N_{attr} * AMBS * sampleCoef_D \tag{2}$$

where, N_{attr} *is the number of attributes in the dataset (i.e. the number of nodes in the underlying Bayesian network).* $AMBS$ *is the average Markov blanket size of the Bayesian network. And* $sampleCoef_D$ *is a data sampling coefficient required to maintain the DAG-faithfulness of extracted sub datasets.*

Algorithm 1. CalculateMSS

Input:
D: Dataset;
ϵ: Threshold;
$mstep$: Maximum loop steps.
Output:
$AMBS$: Average Markov blanket size;
MSS: Minimal sampling size.

```
bestAMBS = 1; step = 0;
sliceSize = 100 * number of attributes in D;
D_sliced = readData(D,nrows = sliceSize); //nrows is the number of rows to read.
BN_DS = LearnBNStructure(D_sliced);
currentAMBS = average Markov Blanket size of BN_DS;
while (currentAMBS > bestAMBS&&step ≤ mstep&&((currentAMBS − bestAMBS) > bestAMBS ∗ ϵ) do
    sliceSize = sliceSize ∗ 2;
    bestAMBS = currentAMBS;
    D_sliced = readData(D, nrows = sliceSize);
    BD_DS = learnBNStructure(D_sliced);
    currentAMBS = average Markov Blanket size of BN_DS;
    step = step + 1;
end while
MSS = number of records in D_sliced;
return bestAMBS and MSS.
```

Theorem 1. *Given a DAG-faithful distribution* P*, there exists two datasets* D_{MSS} *and* D_{S2} *drawn from* P *with sizes* MSS *and* $S2$ *(*$MSS < S2$*) respectively so that the difference of the average Markov blanket size between the Bayesian networks learned from* D_{MSS} *and* D_{S2}*, denoted as* $Diff_{AMBS}(D_{MSS}, D_{S2})$ *is zero. This theorem can be formalized as follows:*

$$\forall P, \exists D_{MSS}, D_{S2}, MSS < S2 | Diff_{AMBS}(D_{MSS}, D_{S2}) = 0 \tag{3}$$

Proof. By Definition 1, D_{MSS} is DAG-faithful. Since $MSS < S2$, D_{S2} is also DAG-faithful. Every DAG-faithful distribution has a unique essential graph [8]. Since D_{MSS} and D_{S2} are drawn from the same distribution P, then the essential graphs of D_{MSS} and D_{S2} are identical. The only difference between an

essential graph and a Bayesian network is the edge direction, but the change of edge direction will not affect the sum of the sizes of Markov blankets. Thus, $Diff_{AMBS}(D_{MSS}, D_{S2}) = 0$.

Based on Eq. (2), to calculate MSS_D, both $AMBS$ and $sampleCoef_D$ are required. But in real life, the network structure is unknown. The only way to estimate $AMBS$ is through learning and obtaining the BN structure. And $sampleCoef_D$ is a varying coefficient dependent on each specific dataset instead of a constant number. To conquer this challenge, in light of Theorem 1, we propose a greedy algorithm called CalculateMSS (Algorithm 1) to calculate MSS.

Algorithm 1 starts with small sub dataset D_{sliced}. It learns the BN from D_{sliced} (Step 4) and obtains average Markov blanket size $AMBS$ (Step 5). Since D_{sliced} may not be DAG-faithful, consequently, BN structure learning algorithms will miss many edges, resulting in small $AMBS$. In order to make D_{sliced} DAG-faithful, the loop in the algorithm (Steps 6-13) doubles $sliceSize$ at each iteration, and stops when $AMBS$ becomes relatively stable. This indicates, based on Theorem 1, that the sub dataset size reaches MSS (Step 14). Algorithm 1 obtains both $AMBS$ and MSS, making $sampleCoef_D$ straightforward to compute using Eq. (2). Section 4.2 will show the experimental results of MSS on three datasets and validate the effectiveness of the algorithm.

3.3 Fast Reservoir Sampling

To reduce the scale of learning task, sub datasets need to be drawn from the whole big dataset for BN learning. To make the sampling more efficient, a novel concept, data reservoir index, is introduced in Definition 2.

Algorithm 2. GetdataReservoirIndex

Input:
$numSubDataset_{MSS}$: Number of sub datasets of size MSS in the whole dataset;
K: Number of sub datasets to be extracted.
Output:
dri: Data reservoir index.
1: Initialize dri as an empty array;
2: **for** i = 1..$numSubDataset_{MSS}$ **do**
3: **if** $i \leq K$ **then**
4: $dri[i] = i$;
5: **else**
6: $removedEntry$ = random(1..i);
7: **if** $removedEnry \leq K$ **then**
8: $dri[removedEntry] = i$;
9: **end if**
10: **end if**
11: **end for**
12: **return** dri.

Definition 2. *(data reservoir index). A data reservoir index, denoted as dri, is an array that contains K elements, and is produced by reservoir sampling of K integers from one to* $numSubDataset_{MSS}$ *where* $numSubDataset_{MSS}$ *is the total number of sub datasets of size MSS in the whole dataset.*

Based on Definition 2, an algorithm named *GetdataReservoirIndex* (Algorithm 2) is proposed. It uses reservoir sampling to obtain *dri*. Since it operates in integer domain up to $numSubDataset_{MSS}$, the computation is very efficient.

After obtaining *dri* and sorting it, K sub datasets can be drawn efficiently from the whole dataset in one pass by extracting data records starting from $dri[i] * MSS$ and ending at $dri[i] * (MSS + 1)$, $i = 1, 2, ..., K$.

3.4 Ensumble Learning

After obtaining the sub datasets, RSEM calls the final procedure *Ensemblelearning* (Algorithm 3) to produce the final BN structure from the big dataset.

Algorithm 3. Ensemblelearning

Input:
D: Dataset ;
D_{sub}: Sub datasets sampled by fast reservoir sampling;
ϵ:Threshold.
Output:
BN_{final}: Final network structure.
1: $BN_{local}[i]$ = LearnBNStructure($D_{sub}[i]$);
2: Obtain the Adjacent Matrix $\mathbf{AM}_i$ from $BN_{local}[i]$;
3: Weight each $BN_{local}[i]$ by BDeu score and transform $BN_{local}[i]$ into a Weighted Adjacent Matrix $\mathbf{WAM}_i$;
4: Sum all $\mathbf{WAM}_i$ using Equation (4) to get the final weighted adjacent matrix **FWAM**;
5: **if** $\mathbf{FWAM}[i, j] > \epsilon$ **then**
6: Set $BN_{final}[i, j] = 1$;
7: **end if**
8: **return** BN_{final}.

The algorithm invokes a BN learning algorithm (e.g. hill climbing) to learn local BN structure for each sub dataset (Step 1). Then, it uses BDeu score [9] to weight these local structures and transform them into weighted adjacent matrix $\mathbf{WAM}_i$, $i = 1, 2, ..., K$ (Step 3). Next, the algorithm sums all $\mathbf{WAM}_i$ using Eq. (4) to obtain the final weighted adjacent matrix **FWAM** (Step 4).

$$\mathbf{FWAM} = \sum_{i=1..K} \mathbf{WAM}_i \tag{4}$$

If an edge exists between node i and node j in majority of local structures, then $\mathbf{FWAM}[i, j]$ should be larger than a threshold ϵ. Therefore, Algorithm 3

adds an edge between i and j in the final network, transforming **FWAM** into the final network structure (Step 5-7).

4 Experiments and Discussion

4.1 Experimental Setup and Datasets

To validate the effectiveness of our proposed method, two experiments were conducted. The first experiment used three synthetic big datasets to confirm the effectiveness of MSS calculation as well as to evaluate the learning accuracy and the computation efficiency of RSEM. The second applied RSEM to a real-world big dataset, in order to show that the method can effectively model causal relationships.

The experiment environment is as follows. The computer is Dell PowerEdge R710, with Intel(R) Xeon(R) CPU E5640, 2.66 GHz, 12 M Cache, and Memory 16 GB (82 GB), 1066 MHz, running the operating system of Windows Server 2008 R2 Enterprise 64-bit, Service Pack 1.

The experiments were run in the R environment (version 3.1.1). Hill climbing and MMHC algorithms [11] in the Bnlearn R Package [19] were used to learn BN structures. The number of sampled sub datasets is 10.

Table 1 lists the datasets (CSV files) used in our experiments. Three synthetic datasets with large data volumes were generated using the data simulation module of the SamIam tool (http://reasoning.cs.ucla.edu/samiam/) from three known Bayesian networks: Child [20], Alarm [21] and HEPAR2 [22]. These known networks provide ground truth for the comparison of average Markov blanket size (AMBS) between the learned networks and the original ones, and for the resulting structural hamming distance (SHD). In Table 1, HMDALAR [23] is a real-world dataset from the Data.gov portal, representing 2009 Home Mortgage Disclosure Act (HMDA) Loan Application Register (LAR) Data.

Table 1. Experimental datasets

Name	#Rows(million)	Size(GB)	#Attributes	Domain
Child	10	1.2	20	Medical
Alarm	10	1.9	37	Weather
HEPAR2	10	4.9	70	Medical
HMDALAR	5.8	3.8	45	Finance

4.2 MSS Cacluation Results

Table 2 shows the computation results for minimal sampling size (MSS) and comparison between calculated AMBS and actual AMBS.

Table 2. MSS and AMBS comparison

Dataset	Calculated MSS	Calculated AMBS	Actual AMBS
Child	4,000	3.00	3.00
Alarm	14,800	3.30	3.51
HEPAR2	224,000	4.29	4.51
HMDALAR	40,000	8.00	n/a

From the first three lines in the table, we observe that the calculated AMBS by Algorithm 1 is close to the actual AMBS, indicating an accurate estimation of the BN complexity.

With the purpose of verifying the correctness of calculated MSS, letting the calculated MSSs (Table 2) be reference values, we used the hill climbing algorithm to perform BN learning on the synthetic datasets by doubly decreasing and increasing the values of MSS, and recorded the resulting SHDs. Figure 2 shows SHD trends over varying MSS on three synthetic datasets.

From the curves in Fig. 2, we observe that SHD rises sharply with the decrease of MSS starting from the reference value, while starting from the calculated MSS (second column in Table 2), SHD becomes stable with the growth of MSS. In other words, the calculated MSS by our algorithm (Algorithm 1) is a reasonable tradeoff between learning accuracy and computational efficiency.

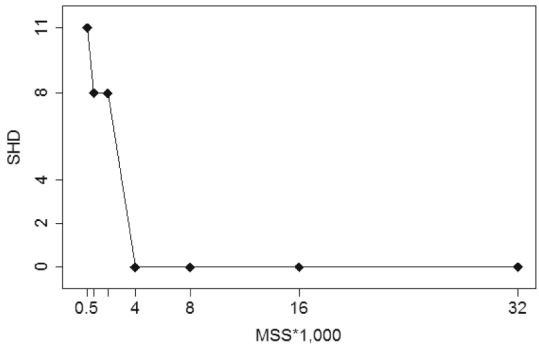
In short, the above experimental results (Table 2 and Fig. 2) confirm the effectiveness of MSS calculation in the proposed RSEM method.

4.3 Results on the Synthetic Datasets

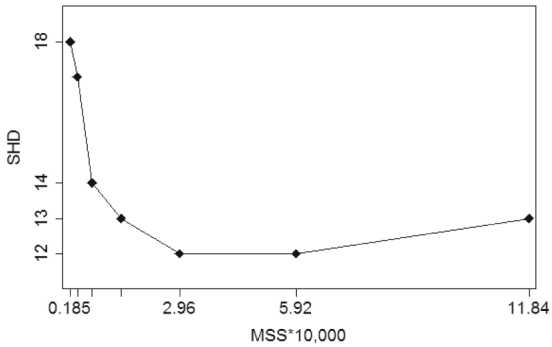
Table 3 shows the comparison of structural hamming distances (SHDs) and computation time for learning BN structures from the datasets between our method (RSEM) and whole dataset learning (WDL) using hill climbing algorithm. When applying RSEM, the threshold ε of the ensemble learning procedure is 0.667.

Table 3. SHD and computation time

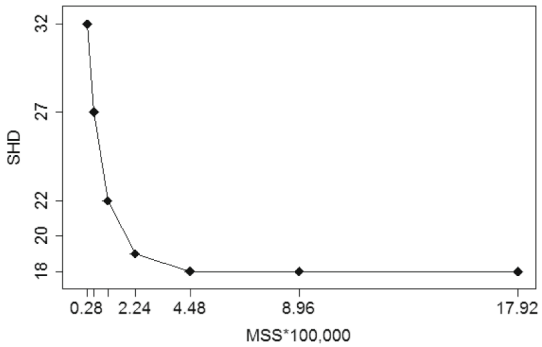
Dataset	Total # of edges in original BNs	SHD		Computation time (minute)	
		RSEM	WDL	RSEM	WDL
Child	25	0	0	1.5	10.1
Alarm	46	13	13	3.20	22.9
HEPAR2	123	17	Failure	34.10	Failure (insufficient memory)
HMDALAR	n/a	n/a	Failure	19.60	Failure (insufficient memory)



(a) Child



(b) Alarm



(c) HEPAR2

Fig. 2. SHD trends over varying MSS on the synthetic datasets.

From the third column of Table 3, we can find that RSEM achieves the same SHD compared with whole dataset learning (WDL) for the Child and Alarm datasets. In particular, RSEM found the correct network for the Child dataset (SHD=0). For the HEPAR2 dataset, RSEM identified over 86 % of the correct edges while WDL failed due to insufficient memory. These results indicate a high learning accuracy of our proposed RSEM.

Regarding the comparison of computation time (the last column in Table 3), it is observed that RSEM achieves nearly an order of magnitude improvement in computation time on the Child and Alarm datasets compared with WDL. Meanwhile, the HEPAR2 and HMDALAR datasets are too big to learn the BN structure from the whole dataset, resulting in computation failure caused by insufficient memory. But our method finished successfully within an hour for both big datasets.

The above experimental results confirm high learning accuracy and good computation efficiency of RSEM.

4.4 Results on the Real-World Dataset

For the HMDALAR dataset, there is no ground truth for the comparison of average Markov blanket size (AMBS) between the learned networks and the original ones, and for the resulting structural hamming distance (SHD). Nonetheless, the following results (cf. Figs. 3 and 4) on the real-world dataset show that our

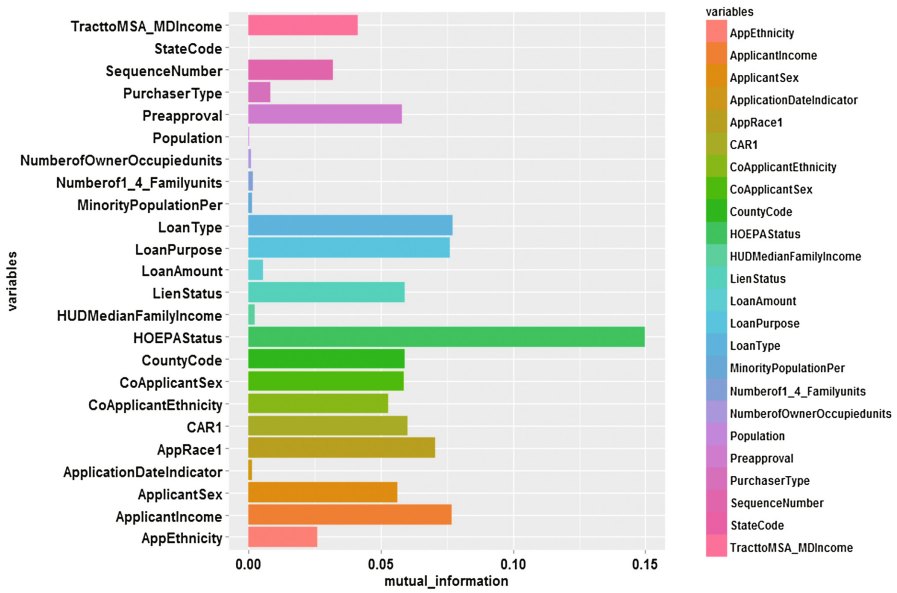


Fig. 3. MI of class variable ActionType and other variables in a sub dataset of HMDALAR

method (RSEM) can sample appropriate sub datasets from the big dataset as well as effectively model causal relationships between the data attributes.

After analyzing the HMDALAR dataset, we found that the ActionType attribute of the data is a class variable. Based on the calculated MSS (40,000) in Table 2, ten sub datasets were sampled from the big dataset. Figure 3 shows the mutual information (MI) of class variable ActionType and other variables in one of the sub datasets. In Fig. 3, we can find that variables HOEPAStatus (Home Owners Equity Protection Act Status), LoanType, ApplicantIncome, LoanPurpose, and AppRace1 (the race of the first applicant) have the top five MI values with the class variable. This is reasonable because from the perspective of loan approval, these variables indeed have a major impact on the approval decision. On the other hand, Fig. 3 indicates that state code, population, numberOfOwnerOccupiedUnits (number of units occupied by the owner), MinorityPolulationPer (Percentage of minority population) have the lowest MI values with the class variable.

As for the modeling of causal relationships between the data attributes, we applied RSEM to the ten sub datasets and produced the final BN. Figure 4 shows the Markov blanket of node ActionType in the Bayesian network. Observing the Markov blanket in Fig. 4, we find that the variables (Preapproval, PurchaserType, HOEPAStatus, and TractooMSA_MDincome) that have direct causal relationships with the class variable are modeled in the Markov blanket. Furthermore, variable Preapproval has six parents including LoanAmount, LoanPurpose, ActionType, Numberof1_4_Familyunits, HUDMedianFamilyIncome, and PurchaserType, which are truly important decision-making factors in loan pre-approval. On the other hand, most variables that have a low

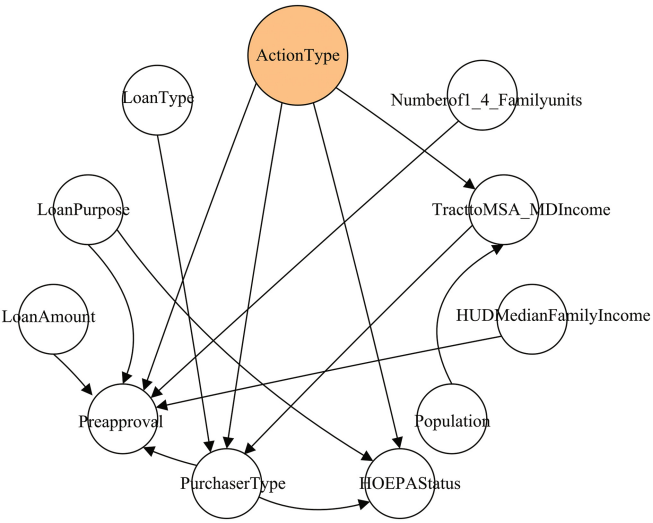


Fig. 4. Markov Blanket of node ActionType in the learned BN from HMDALAR

MI value are not in the Markov blanket of the node ActionType. This shows the effectiveness of RESM in modeling causal relationship for the real world dataset.

5 Conclusion

In this paper, we have proposed a reservoir sampling based ensemble method for Bayesian network structure learning from big data. We have demonstrated through experiments that our method can sample appropriate sub datasets from big datasets using the probabilistic approximation technique, and perform Bayesian network structure learning from big datasets in an accurate and efficient way. This method allows Bayesian network structure learning from big data using a conventional computation platform rather than a big data processing platform. Our future work focuses on enhancing the ensemble method to obtain higher learning accuracy.

Acknowledgments. This work was supported by the Natural Science Foundation of Jiangsu Province, China (Grant No. BK20141420 and Grant No. BK20140857) and the "Six Talent Peaks Program" of Jiangsu Province, China (Grant No. 2008135).

References

1. Ben-Gal, I.: Bayesian Networks. Encyclopedia of Statistics in Quality and Reliability. Wiley, New York (2007)
2. Zhang, Y., Zhang, Y., Swears, N., et al.: Modeling temporal interactions with interval temporal bayesian networks for complex activity recognition. IEEE Trans. Pattern Anal. Mach. Intell. **35**(10), 2468–2483 (2013)
3. Fenton, N.E., Neil, M.: A critique of software defect prediction models. IEEE Trans. Softw. Eng. **25**(5), 675–689 (1999)
4. Sun, S., Zhang, C., Yu, G.: A bayesian network approach to traffic flow forecasting. IEEE Trans. Intell. Trans. Syst. **7**(1), 124–132 (2006)
5. Al-Jarrah, O., Yoo, P., et al.: Efficient machine learning for big data: A review. Big Data Res. **2**(3), 87–93 (2015)
6. Fang, Q., Yue, K., Fu, X., Wu, H., Liu, W.: A mapreduce-based method for learning bayesian network from massive data. In: Ishikawa, Y., Li, J., Wang, W., Zhang, R., Zhang, W. (eds.) APWeb 2013. LNCS, vol. 7808, pp. 697–708. Springer, Heidelberg (2013)
7. Wang, J., Tang, Y., Nguyen, M., Altintas, I.: A scalable data science workflow ap-proach for big data bayesian network learning. In: Proceedings of the 2014 IEEE/ACM International Symposium on Big Data Computing (BDC 2014), pp. 16–25 (2014)
8. Cheng, J., Greiner, R., Kelly, J., Bell, D., Liu, W.: Learning bayesian networks from data: An information-theory based approach. Artif. Intell. **137**(1–2), 43–90 (2002)
9. Heckerman, D., Geiger, D., Chickering, D.: Learning bayesian networks: The combination of knowledge and statistical data. Mach. Learn. **20**, 197–243 (1995)

10. Pearl, J.: Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. Series in Representation and Reasoning. Morgan Kaufmann, San Mateo (1988)
11. Tsamardinos, I., Brown, L.E., Aliferis, C.F.: The max-min hill-climbing bayesian network structure learning algorithm. Mach. Learn. **65**(1), 31–78 (2006)
12. Jiang, L., Li, C., Cai, Z., Zhang, H.: Sampled bayesian network classifiers for class-imbalance and cost-sensitive learning. In: Proceedings of the IEEE 25th International Conference on Tools with Artificial Intelligence (ICTAI), pp. 512–517 (2013)
13. Vitter, J.S.: Random sampling with a reservoir. ACM Trans. Math. Softw. **11**(1), 37–57 (1985)
14. Rokach, L.: Ensemble-based classifiers. Artif. Intell. Rev. **33**(1–2), 1–39 (2010)
15. Hasna, N.J.S.: Weighted ensemble learning of bayesian network for gene regulatory networks. Neurocomputing **150**((B)), 404–416 (2015)
16. Tang, Y., Wang, Y., Cooper, K., Li, L.: Towards big data bayesian network learning - an ensemble learning based approach. In: Proceedings of the IEEE International Congress on Big Data (BigData Congress), pp. 355–357 (2014)
17. Chickering, D., Heckerman, D., Meek, C.: Large-sample learning of bayesian networks is np-hard. J. Mach. Learn. Res. **5**, 1287–1330 (2004)
18. Yoo, C., Ramirez, L., Liuzzi, J.: Big data analysis using modern statistical and machine learning methods in medicine. Int. Neurourol. J. **18**(2), 50–57 (2014)
19. Scutari, M.: Learning bayesian networks with the bnlearn r package. J. Statist. Softw. **35**(3), 1–22 (2010)
20. Spiegelhalter, D., Cowell, R.: Learning in probabilistic expert systems. Bayesian Statistics, 4. Clarendon Press, Oxford (1992)
21. Beinlich, I., Suermondt, H., Chavez, R., Cooper, G.: The alarm monitoring system: A case study with two probabilistic inference techniques for belief networks. In: Proceedings of the 2nd European Conference on Artificial Intelligence in Medicine, pp. 247–256 (1989)
22. Onisko, A.: Probabilistic Causal Models in Medicine: Application to Diagnosis of Liver Disorders. Ph.D. thesis, Institute of Biocybernetics and Biomedical Engineering, Polish Academy of Science, Warsaw (2003)
23. Data.gov - the U.S. Government Open Data: 2009 Home Mortgage Disclosure act (HMDA) Loan Application Register (LAR) Data, Accessed December 15, 2015. http://catalog.data.gov/dataset/2009-home-mortgage-disclosure-act-hmda-loan-application-register-lar-data