Incremental Bayesian network structure learning in high dimensional domains

Amanullah Yasin[1,2] and Philippe Leray[1]
[1]Knowledge and Decision Team,
Laboratoire d'Informatique de Nantes Atlantique (LINA) UMR 6241
Ecole Polytechnique de l'Université de Nantes, France
[2]Balochistan University of IT, Engineering and Management Sciences
Quetta, Pakistan
{amanullah.yasin, philippe.leray}@univ-nantes.fr

Abstract

Recent advances in hardware and software are generating real time and very large data continuously. The learning algorithms need to incorporate novel data. By revising the existing knowledge could be an efficient way to save time and memory. In this paper we proposed an incremental algorithm for Bayesian network structure learning. It could deal with high dimensional domains, where whole dataset is not available and it grows continuously. It learns local structures by limiting search space and use a previous graph as a summary of past data, then performs a constrained greedy hill-climbing search to orient the edges. We evaluated our method on different datasets having several hundreds of variables, in terms of performance and accuracy. The empirical evaluation shows that proposed method is significantly better than state of the art and justifying its effectiveness for incremental use.

1 Introduction

In recent years, there is rapid evolution in hardware and software technologies. Now applications are generating huge amount of data continuously and databases are growing rapidly, for example telecommunication systems, sensor networks, real-time surveillance systems, set of retail chain transactions, etc. In such applications, data is generated on daily basis and it continuously alimenting decision support systems, which have to update their existing knowledge in the light of novel data. Mining such type of data has been an active area of research over the last few years.

The problem becomes more severe when incoming data coming from high dimensional domains, having several hundreds and thousands of variables e.g. biological or social networks.

In traditional batch learning scenario, the whole dataset is available for the learning algorithms. If dataset is growing continuously and we cannot wait for the whole dataset then the obtained model will be outdated with the passage of time. So, revising the model by re-executing batch learning algorithm will take lot of resources to generate decision model. For this reason, incremental learning algorithms are needed in order to efficiently integrate the novel data with the existing knowledge.

Bayesian network structure learning is a powerful tool for graphical representation of the underlying knowledge in data. Bayesian network structure learning is proven as NP-hard problem (Chickering, 1995). Therefore, various heuristics has been proposed for this task. Some of them have been recently developed to learn large-scale networks i.e *Max-Min hill-climbing (MMHC)* (Tsamardinos et al., 2006a), *Bayesian substructure learning* (Nägele et al., 2007), *Local to Global search* (Hwang et al., 2002), *Model based search applied for BN structure learning* (Herscovici and Brock, 2006) and *extension of constraint-based PC algorithm* (Kalisch and

Bühlmann, 2007).

Most of these algorithms use local search heuristic where a local model is build around every single variable. Then, the global structure is learned with the help of these local models. Unfortunately, all these works consider batch learning and need to relearn the model at the arrival of novel data, which is a very costly task.

Regardless of the traditional data mining techniques, incremental Bayesian network structure learning is not a mature field. There are few work addressing this issue i.e. (Roure, 2004; Roure, 2002; Buntine, 1991; Lam and Bacchus, 1994; Friedman and Goldszmidt, 1997; Nielsen and Nielsen, 2008; Zeng et al., 2008). These incremental approaches are mostly based upon traditional "score-and-search" method, which is feasible only for less than hundreds of variables which is not realistic in high dimensional domains.

In this paper we propose a novel method for incremental BN structure learning from high dimensional domains. We adopted an idea of *Max-Min hill-climbing (MMHC)* (Tsamardinos et al., 2006a), one of the most robust structure learning algorithms for high dimensional domains.

This paper is organized as follows: section 2 describes some background. In section 3 we present our incremental method ($iMMHC$) for Bayesian network structure learning. In section 4, experiments varying the $iMMHC$ parameters are presented to show the flexibility of the algorithm and empirical evaluation of $iMMHC$ to justify the effectiveness of the algorithm. Finally, we conclude in section 5 with some proposals for future research.

2 Background

2.1 BN structure learning

Bayesian network (BN) is a graphical representation of a probabilistic relationship among a set of random variables. It represents by a directed acyclic graph (DAG) $\mathcal{G} = (\mathbf{X}, E)$, where $\mathbf{X} = \{X_1, ..., X_n\}$ is the set of nodes or vertices corresponds to the random variables in a graph and the dependencies among variables are expressed with edges E.

Bayesian network structure learning can be consider as a problem of selecting a probabilistic model that explains a given set of data. Without prior knowledge about the network structure, it should be learned from data that best fits the data.

There are three classical approaches often used for BN structure learning: (1) first one consists in detection of (in)dependencies between the variables by performing the conditional independence test on data to find the structure. Later, different assumptions used to direct the edges to get directed acyclic graph (DAG). Its performance is limited with a small number of conditioning set and criticized for complex structures. (2) Second, "score-and-search" based approach is a most widely explored to find the structure of the BN. These methods use a score function to evaluate possible structures and find the network with a maximum score. Different heuristics used to limit the search space and to find the optimum solution. (3) The third "hybrid" approach merges the two previous approaches by first identifying some local structures around target variables, and then proposing one final model by using some global optimization technique constrained with the previous local informations.

These hybrid approaches have mostly been adopted for BN structure learning in high dimensional domains. Let us cite *Max-Min hill-climbing (MMHC)* (Tsamardinos et al., 2006a) and some recent extensions (Nägele et al., 2007; Herscovici and Brock, 2006).

$MMHC$ algorithm first apply "constraint-based" heuristic *Max-Min parent-children (MMPC)* in order to identify local structures, candidate parents-children (CPC) of each target variable. Then it uses a greedy search in order to optimize the global model, starting from an empty graph, with operators *add_edge*, *remove_edge* and *reverse_edge*. Adding an edge $Y \rightarrow X$ can be performed if and only if $Y \in CPC(X)$.

2.2 Incremental BN structure learning

Algorithms presented in this section assume that the data has been sampled from station-

ary domains, or with only small changes in the underlying probability distribution. These algorithms incrementally process data over a *landmark window* (Gama and Gaber, 2007) where window w_i contains data from initial to current time $i * \Delta_w$ where Δ_w is the window size.

(Roure, 2004) proposed two heuristics to transform a batch "score-based" BN structure learning technique (*Hill-climbing search*, HCS) into an incremental one. First heuristic, called *Traversal Operator in Correct Order (TOCO)*, keeps the search path in former learning step. At the arrival of new data $TOCO$ checks the order of the search path. if it is still hold then no need to revise the structure otherwise trigger the HCS to obtain a new model. Second heuristic, called *Reduced Search Space (RSS)*, applies when the current structure needs to be revised. It reduces the search space by considering only high quality models found in the previous search step and avoid to explore low quality models.

(Shi and Tan, 2010a) recently proposed an hybrid method for incremental BN structure learning. They propose two ways to find the local structure, (1) *Tsearch*, using maximum spanning tree ($MWST$) algorithm or (2) using feature selection techniques. The final step, i.e. global optimization, consists in applying a greedy search constrained by these local structures. Incrementality is only managed during this step, by starting the greedy search from the previously obtained BN. This solution is quite limited by not using incrementality during the first step, core of hybrid methods. The same authors proposed an other (very similar) technique (Shi and Tan, 2010b), which is also not feasible for high dimensional domains in incremental environments due to large conditioning set of variables for independence test. The results presented in both works are also using a limited number of variables (maximum of 39 variables).

In contrast to these incremental and hybrid structure learning methods, (Yasin and Leray, 2011) focuses on $iMMPC$ an incremental version of $MMPC$, a "constraint-based" local discovery algorithm, by using $TOCO$ and RSS heuristics. $iMMPC$ learns a local structure around a single target variable by using the

Algorithm 1 iMMHC(D_w)

Require: Data of time window w_i (D_{w_i}), previous top K best neighbors for each variable ($\mathcal{B}$), previous BN structure (BN_{i-1})
Ensure: BN structure (BN_i)

% incremental local identification
for all $X \in \mathbf{X}$ **do**
  $CPC(X)$=iMMPC($X, D_{w_i}, \mathcal{B}$)
end for
% Incremental greedy search
starting model $\mathcal{M}$ = empty graph **or** BN_{i-1}.
Only try operators add_edge $Y \rightarrow X$ if $Y \in CPC(X)$
(no constraint for remove_edge or reverse_edge)
and return BN_i the highest scoring DAG found

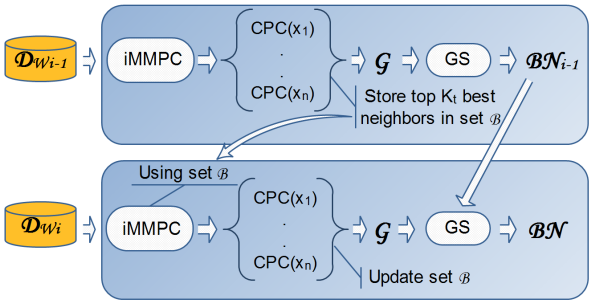

Figure 1: iMMHC outline : dependencies between $iMMHC$ execution for window w_i and previous results.

previous knowledge, and avoids exploring those part of the search space which are previously found with low quality. For this purpose, it maintains a list of search paths (order of the variables included in the candidate parent-children "CPC" set) and top K best variables that have more chance to be consider in CPC.

3 Incremental MMHC

As seen in the previous section, incremental and hybrid BN structure learning algorithm $Tsearch$ uses a "poor' local identification solution and deal with incrementality only during the second phase. In the opposite, $iMMPC$ is only dedicated to propose an incremental solution for the local identification phase.

The main idea of our work is proposing an incremental and hybrid BN structure learning algorithm dealing with incrementality during both phases (local structure identification and global optimization). Our proposal, *incremental*

Benchmark	#Vars	#Edges	Card.	#Inst.	Degree
Alarm	37	46	2 - 4	20,000	6
Barley	48	84	2 - 67	20,000	8
Hailfinder	56	66	2 - 11	20,000	17
Pathfinder	109	195	2 - 63	20,000	108
Andes	223	338	2	10,000	12
Link	724	1125	2 - 4	10,000	17
Gene	801	972	3 - 5	5,000	11
Alarm28	1036	1582	2 - 4	10,000	8

Table 1: Name of the benchmark, number of variables, number of edges, cardinality and degree of the graph used for our experiments.

MMHC (*iMMHC*) is also a two phases hybrid algorithm described in Algorithm 1. In the first step, it discovers the possible skeleton (undirected graph) of the network for a given window, by using *incremental MMPC* (Yasin and Leray, 2011) method. Then a greedy search is initiated. A naive application of *MMHC* would start this greedy search from the empty graph. As proposed by *Tsearch*, we choose in this incremental algorithm to start from the graph learned in the previous time window. This initialization considers adding the edges discovered in the new skeleton but also removing the outdated edges. By this way, *iMMHC* keeps the sense of incremental learning in both phases, by considering the previously learned model and revising the existing structure in the light of new data, as summarized in Figure 1.

4 Empirical evaluation

We carry out several experiments comparing *iMMHC* with the most recent state of the art. Our goals are to evaluate its ability to deal with high dimensional data (up to hundreds of variables) and characterizing the situation where it out performs the other algorithms.

4.1 Experimental protocol

Benchmarks: We have chosen seven well-known networks from GeNIe and SMILE network repository[1]. Numerical characeristics of these benchmarks are summarized in Table 1.

To test the performance in high dimensional data, we also took five "Gene" datasets from MMHC source website[2]. To generate network with thousands of variables, we used BN tiling (Tsamardinos et al., 2006b) method (implemented in Causal Explorer[3]) and generated "Alarm28" network with 1036 variables by tiling 28 copies of Alarm network.

Five datasets are generated from each network. For an incremental purpose, we feed the algorithm in the form of windows of size $\Delta_w =$ 1000, 2000 and 4000.

Algorithms: We compared the both initializations of proposed *iMMHC* algorithm, empty graph and previous DAG denoted by $iMMHC_{\varnothing}$ and $iMMHC_G$ respectively, with the batch *MMHC* (without TABU search) and with the incremental *TSearch* described in section 2.2. We have implemented the original algorithms as described in their articles, in C++ using Boost graph[4] and ProBT[5] libraries.

Independence is measured with the help of Mutual Information. Confidence level of statistical tests is set to 0.05.

4.2 Evaluation measures

We used two metrics to evaluate our algorithm in terms of *computational efficiency* and *model accuracy*. The main task in the algorithm is to compute score function. The complexity of the algorithm depends upon the number of total score *function calls* (i.e. for hybrid learning it is equal to MI calls for independence tests and local score function calls during the greedy search). We remind the readers that the skeleton discovery phase of *iMMHC* algorithm use constraint based approach which already has a less complexity than score-and-search based techniques.

For *model accuracy*, the Structural Hamming Distance (SHD) (Tsamardinos et al., 2006a) is used. SHD compares the learned and original network in terms of structural differences that cannot be statistically distinguished.

[1] http://genie.sis.pitt.edu/networks.html

[2] http://www.dsl-lag.org/supplements/mmhc_paper/mmhc_index.html

[3] http://www.dsl-lab.org/causal_explorer/

[4] http://www.boost.org/

[5] http://www.probayes.com/index.php

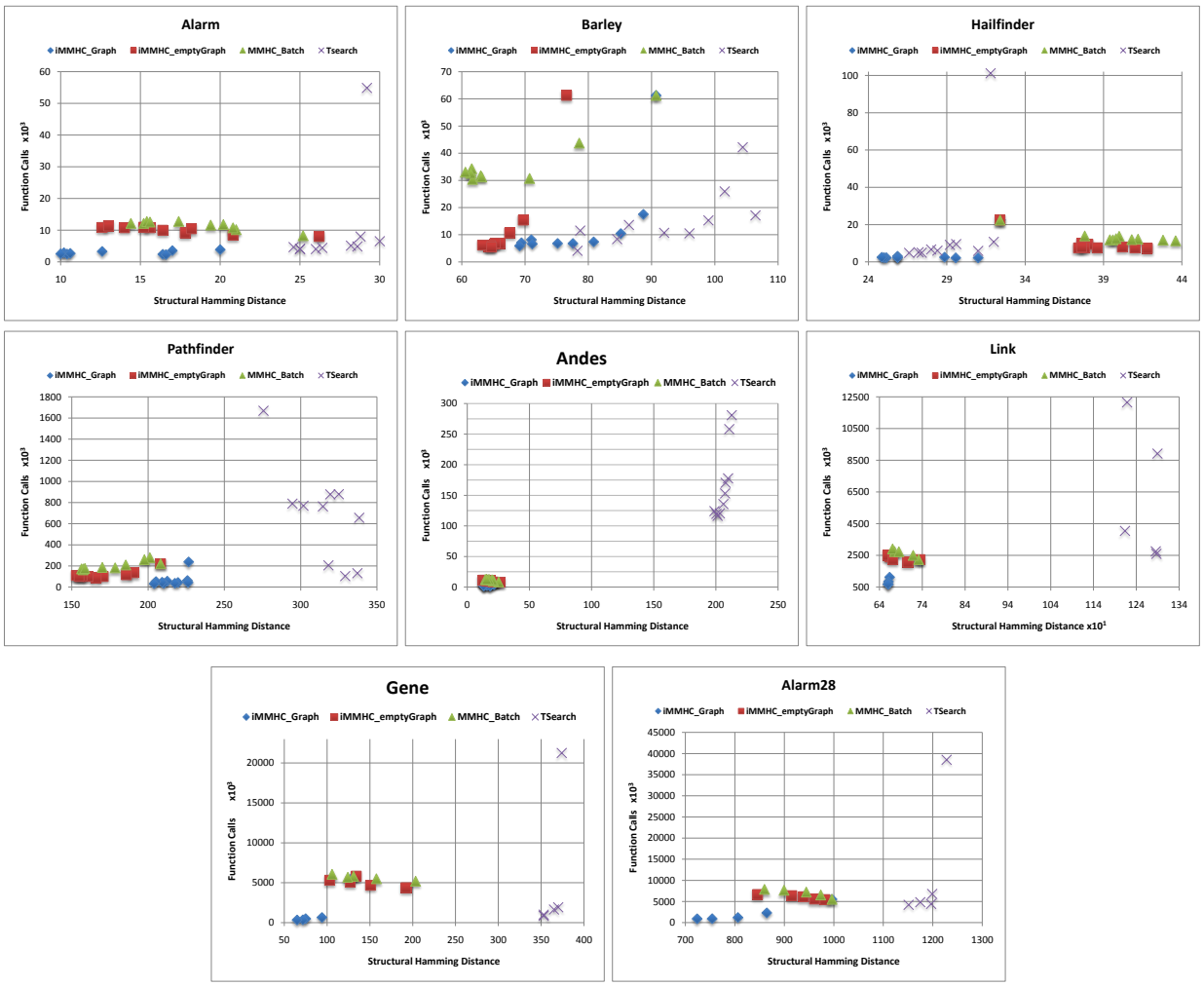

Figure 2: Comparisons of Structural Hamming distance (SHD) and function calls between $MMHC, iMMHC_{\varnothing}, iMMHC_G$ and $TSearch$ algorithms using window of size 2000 (1000 for "Gene")

4.3 Results and discussion

Figure 2 describes the total number of function calls and SHD for each time window and for all the algorithms ($iMMHC_G$, $iMMHC_{\varnothing}$, batch $MMHC$ and $Tsearch$).

***Best parameters*:** We have analyzed two types of initializations (using previous and empty graph) for the greedy search used in $iMMHC$ edge orientation phase. We can observe from Figure 2 that the computational complexity of $iMMHC_G$ is very low compared to $iMMHC_{\varnothing}$, with an higher accuracy except for datasets having a high cardinality ("Barley " and "Pathfinder"). (Friedman and Goldszmidt, 1997) claims that using an existing structure as a summary of past data strongly bias the learning procedure towards this structure. Consequently, after some iterations, this approach locks itself into a particular structure and stops adopting new data. But in our approach we can see in figures 3 and 4 that the quality of the learned model always increases when we considered the previous graph as a summary of past data.

Parameter K is used in $iMMPC$ to store top K most associated variables with the target variable for later use in novel data, because they have more chance to become parent or children

of a target variable. The behavior of $iMMHC$ with different K values is shown in figure 5. We can see that a better accuracy can be obtained when K is more close to the degree of the theoretical graph. Logically, the complexity linearly increases with respect to K. In real world applications, the degree of the underlying graph is not a prior knowledge. As a comparison, usual structure learning algorithms have to limit the maximum number of parents in order to be scalable. Our algorithm only has to limit the number of neighbors to be store in the cache, which is independent of the number of parents of the final DAG. We are then able to control the scalability of our algorithm without controlling the complexity of the final model.

iMMHC for incremental learning: In general, Figure 2 shows that $iMMHC_G$ outperforms $TSearch$ with respect to complexity and accuracy, except in "Hailfinder" where results are similar.

During $Tsearch$ learning, if the skeleton discovery phase ($MWST$) contains lot of false positive edges then these errors propagate to the final structure. As this structure is also used as an input for the next time window, so local errors in skeleton discovery will mislead the incremental structure learning process. An other consequence of false positive edges, the complexity of the global optimization phase is also increased.

$iMMHC$ use an incremental adaptation of $MMPC$ algorithm for this skeleton discovery phase. This algorithm has been proven as a robust solution, limiting the number of false positives. In this context, $iMMHC$ is a more robust algorithm than $Tsearch$.

Incremental versus batch MMHC: Our experimental results in Figure 2 shows that our incremental approach can obtain better quality with lower complexity than the batch original algorithm except the datasets having high cardinality ("Barley " and "Pathfinder"). This result is consistent with Roure's work in (Roure, 2004). With respect to the high dimension of the search space, incremental algorithms can avoid being trapped in some local optima as could be their batch counterparts.

iMMHC for high dimensional domains:

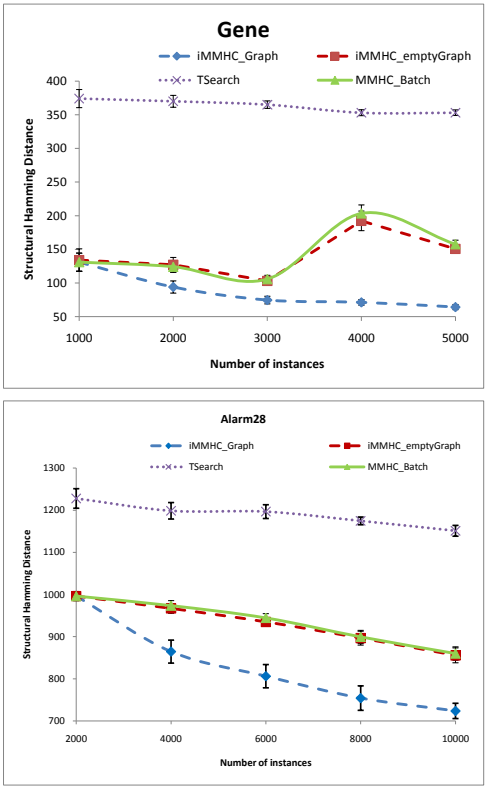

Figure 3: Comparison of model accuracy over incremental learning process for "Gene" and "Alarm28" (average and standard deviation)

To check the ability of the algorithm to deal with high dimensional domains, tests were conducted with "Alarm28 " benchmark (1028 variables). We observed that the results in Figure 2 are much better than batch $MMHC$. $iMMHC$ is an incremental algorithm where the first iteration has the same complexity as the batch algorithm but in the succeeding iterations, this complexity rapidly decreases. For this reason, $iMMHC$ can be adopted for several thousands of variables in incremental environment.

5 Conclusions and future work

We have presented an incremental approach $iMMHC$ for BN structure learning in high dimensional domains where the whole data set is not available. $iMMHC$ uses (1) our $iMMPC$ algorithm for incremental local structure identification and (2) a greedy search for global model optimization, constrained by these local infor-

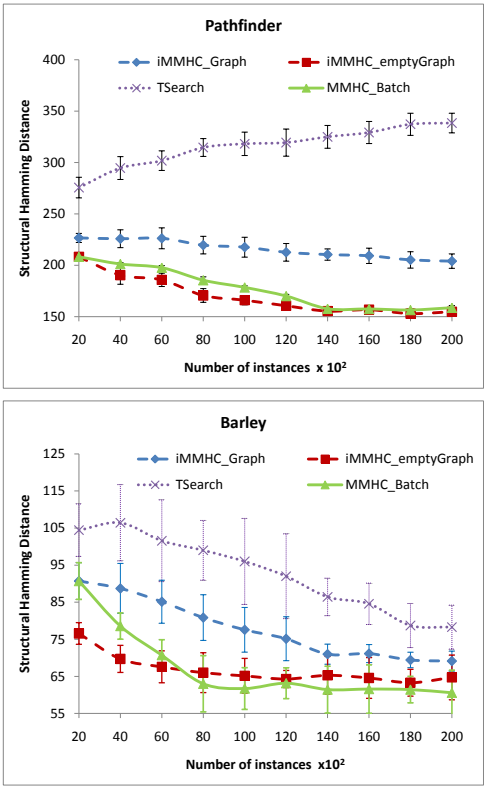

Figure 4: Comparison of model accuracy over incremental learning process for "Barley" and "Pathfinder" (average and standard deviation)

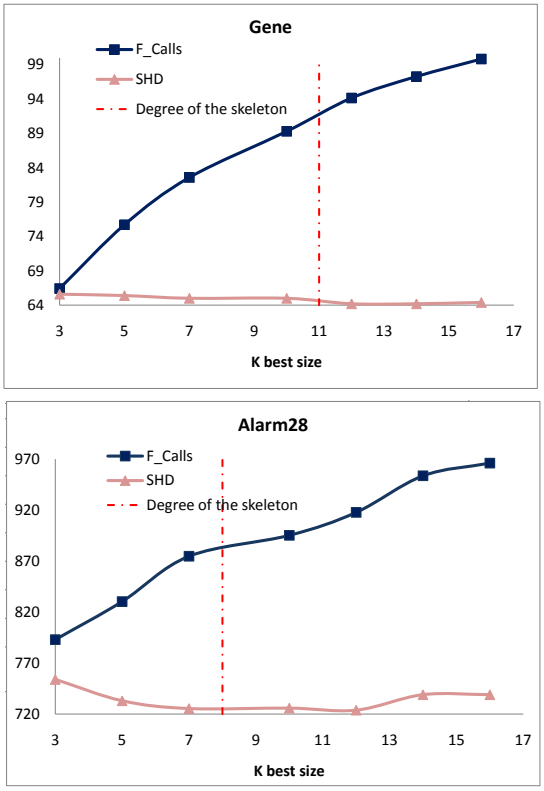

Figure 5: Performance of iMMHC algorithm for different K values (Function calls are divided by 4×10^3 for "Gene")

mations (such as all hybrid approaches) and starting from the previously obtained model (for incremental purpose). By this way, we are able to take into account the previous knowledge in order to reduce the search space for incremental learning. Our experimental results illustrate the good behavior of $iMMHC_G$ compared to a similar incremental algorithm ($Tsearch$) and the batch original one ($MMHC$). So $iMMHC$ could also be an interesting alternative to batch learning for large databases.

Mimicking the original $MMHC$, the second phase of $iMMHC$ adopts a classical greedy search which is the most time-consuming part of the algorithm. So, we could take into account recent extensions of $MMHC$ which improve this phase (Nägele et al., 2007).

$iMMHC$, but also similar BN incremental learning, learns over landmark windows. This is not sufficient for real applications such as data stream mining. One immediate perspective is the adaptation of $iMMHC$ in order to deal with *sliding windows* for unbounded data streams by keeping the less possible information about past data and dealing with non-stationarity. In this context, storing sufficient statistics of past data with ADtrees (Anderson and Moore, 1998) can be possible, with a sequential update of these models as proposed in (Roure and Moore, 2006). Using forgetting coefficient (Anagnostopoulos et al., 2009) is also a first solution for taking into account non-stationarity.

References

Christoforos Anagnostopoulos, Dimitris K. Tasoulis, Niall M. Adams, and David J. Hand. 2009. Temporally adaptive estimation of logistic classifiers on data streams. *Adv. Data Analysis and Classification*, 3(3):243–261.

Brigham Anderson and Andrew Moore. 1998. Adtrees for fast counting and for fast learning of association rules. In *Proceedings Fourth International Conference on Knowledge Discovery and Data Mining*, pages 134–138. ACM Press.

Wray Buntine. 1991. Theory refinement on Bayesian networks. In *Proceedings of the seventh conference (1991) on Uncertainty in artificial intelligence*, pages 52–60, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

David M. Chickering. 1995. Learning Bayesian networks is NP-complete. In *Proceedings of AI and Statistics, 1995.*, pages 121–130.

Nir Friedman and Moises Goldszmidt. 1997. Sequential update of Bayesian network structure. In *Proc. 13th Conference on Uncertainty in Artificial Intelligence (UAI 97)*, pages 165–174.

Joao Gama and Mohamed Gaber. 2007. *Learning from Data Streams – Processing techniques in Sensor Networks.* Springer.

Avi Herscovici and Oliver Brock. 2006. Improving high-dimensional Bayesian network structure learning by exploiting search space information. Technical report, Department of Computer Science, University of Massachusetts Amherst.

Kyu Baek Hwang, Jae Won Lee, Seung-Woo Chung, and Byoung-Tak Zhang. 2002. Construction of large-scale Bayesian networks by local to global search. In *Proceedings of the 7th Pacific Rim International Conference on Artificial Intelligence: Trends in Artificial Intelligence*, PRICAI '02, pages 375–384, London, UK. Springer-Verlag.

Markus Kalisch and Peter Bühlmann. 2007. Estimating high-dimensional directed acyclic graphs with the pc-algorithm. *J. Mach. Learn. Res.*, 8:613–636, May.

Wai Lam and Fahiem Bacchus. 1994. Using new data to refine a Bayesian network. In *Proceedings of the Tenth Conference on Uncertainty in Artificial Intelligence*, pages 383–390.

Andreas Nägele, Mathäus Dejori, and Martin Stetter. 2007. Bayesian substructure learning - approximate learning of very large network structures. In Joost Kok, Jacek Koronacki, Raomon Mantaras, Stan Matwin, Dunja Mladenic, and Andrzej Skowron, editors, *Machine Learning: ECML 2007*, volume 4701 of *Lecture Notes in Computer Science*, pages 238–249. Springer Berlin / Heidelberg.

Søren Holbech Nielsen and Thomas D. Nielsen. 2008. Adapting Bayes network structures to non-stationary domains. *Int. J. Approx. Reasoning*, 49(2):379–397.

Josep Roure and Andrew W. Moore. 2006. Sequential update of ADtrees. In *ICML '06: Proceedings of the 23rd international conference on Machine learning*, pages 769–776, New York, NY, USA.

Josep Roure. 2002. An incremental algorithm for Tree-shaped Bayesian network learning. In *Proceedings of the 15th European Conference on Artificail Intelligence*, pages 350–354. IOS Press.

Josep Roure. 2004. Incremental hill-climbing search applied to Bayesian network structure learning. In *First International Workshop on Knowledge Discovery in Data Streams. (KDDS-ECML).*

Da Shi and Shaohua Tan. 2010a. Hybrid incremental learning algorithms for Bayesian network structures. In *Proc. 9th IEEE Int Cognitive Informatics (ICCI) Conf*, pages 345–352.

Da Shi and Shaohua Tan. 2010b. Incremental learning bayesian network structures efficiently. In *Proc. 11th Int Control Automation Robotics & Vision (ICARCV) Conf*, pages 1719–1724.

Ioannis Tsamardinos, Laura E. Brown, and Constantin F. Aliferis. 2006a. The max-min hill-climbing Bayesian network structure learning algorithm. In *Machine Learning*, pages 31–78. Springer Netherlands, Inc.

Ioannis Tsamardinos, Alexander R. Statnikov, Laura E. Brown, and Constantin F. Aliferis. 2006b. Generating Realistic Large Bayesian Networks by Tiling. In *Proceedings of the Nineteenth International Florida Artificial Intelligence Research Society Conference*, pages 592–597, California, USA. AAAI Press.

Amanullah Yasin and Philippe Leray. 2011. iMMPC: a local search approach for incremental Bayesian network structure learning. In *Proceedings of the 10th international conference on Advances in intelligent data analysis X*, IDA'11, pages 401–412, Berlin, Heidelberg. Springer-Verlag.

Yifeng Zeng, Yanping Xiang, and S. Pacekajus. 2008. Refinement of Bayesian network structures upon new data. In *The 2008 IEEE International Conference on Granular Computing, GrC 2008, Hangzhou, China, 26-28 August 2008*, pages 772–777. IEEE.