Faster Bayesian Context Inference by Using Dynamic Value Ranges

Korbinian Frank and Patrick Robertson
Institute of Communications and Navigation
German Aerospace Center (DLR)
82234 Oberpfaffenhofen, Germany
Email: {korbinian.frank | patrick.robertson}@dlr.de

Sergio Fortes Rodriguez and Raquel Barco Moreno
E.T.S. de Ingeniería de Telecomunicación
University of Málaga
29071 Málaga, Spain
Email: {sergio.fortes.rodriguez@ieee.org | rbarco@uma.es}

***Abstract*—This paper shows how to reduce evaluation time for context inference. Probabilistic Context Inference has proven to be a good representation of the physical reality with uncertain or missing information, giving with the probability also a measure of the quality of information. As the inference complexity is very high, the complexity of the the to be evaluated rule (representing a share of the real world) should be reduced as far as possible. Therefore we present an approach to select only relevant values of context types and to adapt this selection during its usage time. A short proof of concept indicates that both targets, reducing inference time and maintaining quality of information, can be reached with the proposed approach.**

***Keywords*-Context Inference, Bayesian Inference, Dynamic Value Ranges, Bayeslets**

I. Introduction

A. Motivation

The utility of services in context aware frameworks is depending to a large extent on the information it processes. This information is context-aware and can stem from a huge number of different providers, either services themselves or sensors. They all have different relevancy to the user, different accuracy and different confidence levels, which therefore should be provided to the information consumer. This problem holds in particular for *context inference*, where conclusions are drawn from the available information and the uncertainty about the input information impacts the uncertainty about the conclusion.

Bayesian Networks (BN, see [1]) are using conditional probabilities as a means to represent the quality of information and are more and more used for inference in different research fields, such as medical research, genetics, insurance analysis, and fault handling. A BN encompasses a set of *random variables (RV)* that represent the domain of interest and encodes the important relationships between these variables, such as causality and statistical dependence and in-dependence, by directed edges. Discrete RVs encompass a set of mutual exclusive *values* (also called *states*). The transition probabilities from all causes for every state of the RV are represented in the *Conditional Probability Distribution (CPD)*.

Processing BNs however is NP hard in the number of nodes [2]. This comes from the amount and size of the CPD tables that grow exponentially with the number of nodes, states per nodes and number of incoming edges. This evaluation complexity is fatal if BNs are exceeding a certain size.

If CPD tables however are kept small, inference, i.e. processing of these tables, is much faster. Therefore the number of necessary conditional probabilities in the tables has to be reduced. The size of a CPD table of a RV A, $\overline{T}_A$, is given by $\overline{T}_A = |A| \cdot \prod_{J \in \mathbf{S}_i} |J|$, $\mathbf{S}_i$ being the set of parent RVs of A and $|J|$ being the number of states of J. It is limited by the following borders: $2^{|\mathbf{S}_i|+1} \leq \overline{T}_A \leq k^{|\mathbf{S}_i|+1}$, where $k = \max_{J \in \mathbf{S}_i \cup \{A\}} |J|$. Minimizing a *value range (VR)*, the reduction of any $|J|$ and thereby k, is hence desirable to improve evaluation time. The challenge here however is, that all reduced nodes J' with $|J'| \leq |J|$ however still have to contain all necessary, i.e. relevant states.

The relevant states however depend again very much on context: different users may judge different states more or less important, for themselves or other people's context information. They also may vary in time, as in the user's location. This leads us to the need, not only to reduce the VR, but to modify them dynamically based on current requirements.

The aim of this paper is to present a methodology to reduce dynamically the value ranges of random variables in Bayesian networks for context inference in ubiquitous computing frameworks. With this reduction, real time inference should be made possible while not reducing the quality of information of the inferred information. We will present the process, the used theory and give some first evaluation results.

II. Related work

The issue of discretization or *repartitioning* has been developed in a number of works. Most methods use discretization of continuous values for *classification learning*.

That is the case of Fayyad in [3], where a *Entropy Minimum Description Length Principle* is used to select recursively the thresholds to discretize continuous values in a top-down way. Their work establishes a Minimum Descriptive Length Principle Criterion to stop this recursion.

For Bayesian applications, Barco et al. applied part of these concepts to implement diagnosis in mobile communication networks [4]. They analyse different techniques and the performance of using discretized value ranges.

Other projects use discretization after the construction of the network and the selection of parameters. In this way the work of Clarke and Burton in [5] contains methods for repartitioning prior to (*initial partitioning*) or during the construction of a BN, which they call *dynamic repartitioning*. Repartitioning follows a bottom-up approach, where the two most appropriate states to be merged are selected based on some entropy criteria. The merging process is applied then recursively to the current value range obtained from the last merging step. They also establish a stopping criterion for the recursion based on a concept, called *knee point*, that we will present in detail in section III-A. The knee point is applicable to both metrics that are used by Clarke and Burton, Entropy and the *Minimum Description Length (MDL)* which was also used by Bouckaert in [6].

Other works in the Bayesian field also aim for the discretization during learning, like for instance the paper by Friedman and Goldsmith [7] using the MDL score.

A *dynamic anytime discretization* process is developed by Kozlov and Koller [8], for hybrid Bayesian networks. For this they make use of the *Kullback-Leibler* (and its weighted versions) distance between different probability density functions.

III. Dynamic Repartitioning

This section is dealing with the repartitioning process in already constructed, discrete Bayesian networks. The proposed methodology will follow a bottom-up approach, where initially each variable begins with its complete VR which may contain a high number of values. Subsequently, the number of values will be reduced by merging values.

Thereby repartitioning uses different methods and criteria to minimize information loss and computational cost of the repartition process. Along all these methods probabilistic information contained in the Bayesian network will be used, following the scheme in Figure 1. The original VRs are subject to a process called *State Selection*, where the values that should be merged with others are selected. The merging is executed recursively until a stopping criterion is met. Once a VR is simplified it still has to monitor changing conditions like context or current evidence, which might trigger a new repartitioning process.

The first subsection is dedicated to criteria and methodologies to select the states to merge and the recursion depth. Afterwards in subsection III-B, methods to improve the process by inclusion of human knowledge or services requirements will be discussed, extending these methods also in the following subsection. In subsection III-D the methods developed to manage exchange of information between different value ranges are described, before the last subsection III-E explains how new repartition processes would be triggered.

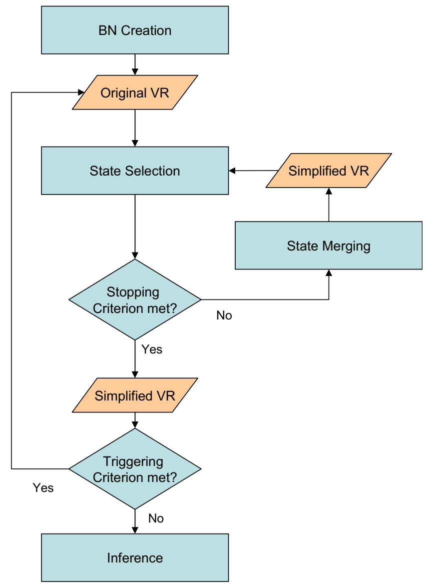

Figure 1. Steps of a Dynamic Repartitioning System

A. State Selection and Stopping Point Criteria

Often in our approach the sequence of actions to guide the repartitioning is based on a recursive process that results in the most suitable states to merge. The applied rules are called *state selection criteria*. Each time in the recursive process the system receives a new (modified) set of candidates to be merged, it has to decide based on certain *stopping point criteria*, if these values must be merged or if the repartitioning process should stop.

Originally these methods were developed for discretization processes for *data batches* in the literature. In this case the expressions depend on the number of *cases* or sets. Since our methods are applied to already constructed Bayesian networks with discrete variables, probabilistic data (conditional probabilities, prior marginalization, ...) should be directly utilized by these criteria.

In the present work, some of the most common and characteristic discretization methods have been adapted to our conditions, as follows:

Equal Number of States: Equal Number of States (ENS) [9] divides the VR into subsets that take each n adjacent values and then merges all states of each subset. This method however does not take into account the probabilistic information of the network. This could be useful only for numerical states, like e.g. temperature, in order to reduce precision and therefore overhead.

Temperature	Degrees						
original	1	2	3	4	5	6	...
ENS processed	1-2		3-4		5-6		...

The quality of this merging depends largely on the probabilistic effects of the values to be merged. If they cause different behaviour in a connected node, loss of information by merging would be quite high. Another prerequisite for it to make sense is the kind of information encoded in the states. For nominal values, e.g. colours, the ordering might not be of any importance, while for numerical values, ordering usually does make sense. If ENS was chosen for clustering, the RV would consequently have to carry information about the type of its values.

To select the number of states to merge some criterion of minimum or ideal number of states for each node to maintain a certain precision or computational cost may be established and the merging process would be adapted to accomplish it. Existing database clustering methods (e.g. density-based clustering) could be used to determine the states to be merged.

Entropy Minimization Discretization: Entropy Minimization Discretization (EMD) adapts the method of [3], using the next Entropy expression for any value range subset, V_S, of V:

$$Ent(\mathbf{V_S}) = -\sum_{v \in \mathbf{V_S}} P(V=v)log(P(V=v)) \qquad (1)$$

Based on this the *class information entropy induced by the partition of V in two subsets* $\mathbf{V_1}$ *and* $\mathbf{V_2}$ is defined as:

$$Ent(\mathbf{V_1}, \mathbf{V_2}) = \frac{|\mathbf{V_1}|}{|V|} Ent(\mathbf{V_1}) + \frac{|\mathbf{V_2}|}{|V|} Ent(\mathbf{V_2}) \qquad (2)$$

The optimal partition is determined selecting the subsets $\mathbf{V_1}$ and $\mathbf{V_2}$ with the minimal $Ent(\mathbf{V_1}, \mathbf{V_2})$ amongst all the candidates. Merging all the states of each subset provides a binary discretization.

Minimum Merged Probability: Minimum Merged Probability (MMP) tries to minimize the loss of entropy caused by repartitioning and is based on the concepts exposed by Clarke and Burton in [5], using the information given by the prior probabilities of each state. With the formula of the entropy given in Equation 1, it is possible to prove the following: If the states v_i and v_{i+1} selected to be merged are chosen such that the merged probability, $p(v_i)+p(v_{i+1})$ or the probability of the resulting state is minimized, the reduction of the entropy is minimum as well. Therefore these states with the minimum merged probability will be selected.

As stopping criterion, the *knee point* of the entropy function over the number of partitions is used. This point is where the change in the entropy becomes greater than the reduction in the number of states. It is calculated as where the next expression begins to decrease.

$$k_{max} \cdot Ent(p_k) \; - \; k \cdot Ent(p_{k_{max}}), \qquad (3)$$

k being the current number of states, k_{max} the initial value and $Ent(p_k)$ the VR entropy for these k states.

Another feature of this method is the possibility to allow merging of only adjacent values or non-adjacent values. Processing only continguous nodes is indispensable for the case of numerical or certain categorical values. For other value ranges however, selecting and processing the most suitable states regardless of their relative position within the VR, would give a better repartitioning result.

B. Service personalization of the repartitioning process

The final objective of our system is to obtain probabilistic information about different values of certain variables of the network. We will call these nodes *nodes of interest* or *service nodes*, because very often the information for our application is contained only in certain nodes of the network. That is the reason to apply our repartitioning methods to these nodes, in order to decrease the computational cost of the Bayesian processes.

As these concepts are part only of the human knowledge and the system should be capable of automatic discretization only based on the probabilistic nature of the network, we established two ways to introduce this knowledge in order to guide the discretization and thereby to increase its quality.

- *Clustering:* As stated before, sets of states with very different probabilistic behaviour that remains undetected by the states selection criteria, still could have a strong human-knowledge relation. To allow the inclusion of this instruction we can establish different sets (or *Clusters*) in a value range, where the merging process is only realized for states of the same set. For instance, in node `Activity` of Figure 2 the values about activities outside the workplace could belong to the same cluster.
- *Protected States:* or *unmergeable states* are the values of a RV that must not be merged with others. Hence, we can avoid undesirable decreases of precision in the most interesting values.

As their name indicates, the service nodes or the desired unmergeable states are application dependent. In pervasive computing we can imagine a situation like the one shown in Figure 2. Two different services require probabilistic information from an inference engine indicating their different states of interest of some context variables. The inference engine then should generate two new Bayesian networks from the original. These BNs will have different VRs in order to provide the inferred information with maximum performance and minimum loss of accuracy in the range of interest of each service.

C. Protection Extension

The use of any of the discretization methods, as Figure 3 in the evaluation section will show, provides reduced quality in the inference process introducing high variance with

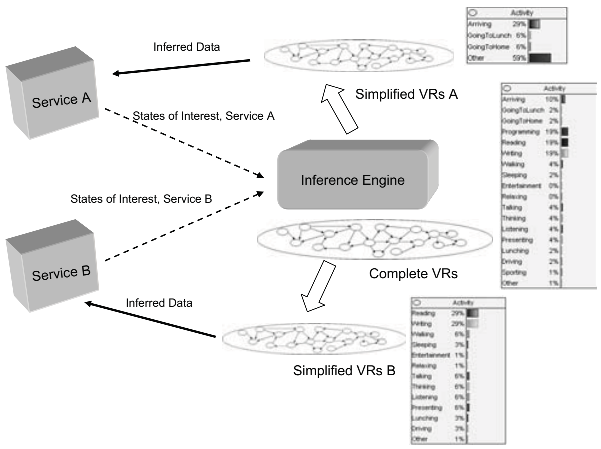

Figure 2. Distributed Bayesian environment with different services requesting probabilistic information from a probabilistic server

respect to the original system, even in the set of protected values. To increase the quality by the establishment of states of interest, it is necessary to extend this protection to the most related values of other nodes. To allow that, the value of the *partial mutual information* between the states of interest and the states of the connected nodes has to be taken into account. Therefore we defined based on [7] the following function, that calculates the partial mutual information for a set of interest $\mathbf{y_s}$ for the node Y and one state of its parent $X = x$.

$$I(Y = \mathbf{y_s}, X = x) = \sum_{y \in \mathbf{y_s}} p(x,y) \left| log(\frac{p(x,y)}{p(x)p(y)}) \right| \quad (4)$$

The absolute value was applied in order to maintain the non-negative characteristic of the information, since the characteristic of $H(X,Y)$ is lost due to taking only a subset of states. Hence, the term $\left| log(\frac{p(x,y)}{p(x)p(y)}) \right|$ provides information about how dependent the states are. $p(x,y)$ weights the expression based on the joint probability of the two states. But also other weighted measures could be adopted [8].

For the mutual information between a child's state $Y = y$ with a states-of-interest set of its parent's state $\mathbf{x_s}$, the equation is:

$$I(Y = y, X = \mathbf{x_s}) = \sum_{x \in \mathbf{x_s}} p(x,y) \left| log(\frac{p(x,y)}{p(x)p(y)}) \right| \quad (5)$$

Therefore, if there is a states-of-interest set in one node, it is possible to calculate this measure for each state of the nodes related and extend the protection to those with the highest mutual information. If the protection extension is applied recursively to all related nodes with high mutual information, it could be extended to all the BN.

This allows a high improvement in the quality of the probabilistic information given by the BN in the values of interest. Moreover, the *protection extension (PE)*, can be adjusted to be extended in a certain number of steps. Establishing a boundary b, a minimum for the mutual information, the protected states can be chosen automatically. If $I > b$ the state is protected, otherwise not. If b is chosen small, more states will be protected, which leads to higher precision in the inferences at the cost of less reduction in the number of network states. For the remaining, unprotected states, any of the previous repartitioning methods can be applied, i.e. all unprotected states can be merged to one state.

This was the option applied for the evaluation section IV.

D. Interface between different Value Ranges

In real applications, the service execution will be based on probabilistic information given by the inference networks. Although the methodology of the previous subsections implies influence of each service in the repartitioning by clustering and PE, this could only appear at the beginning of the inference usage in a kind of configuration stage. Besides, the value range processing must be transparent for the applications supported. This implies that the inference BN and the application that provides *evidence* must share a common VR (or at least part of it), to allow the exchange of information. Different solutions have been developed by us for each case:

- *Applying Evidence:* The case of hard evidence is very easy to solve. Saving the relation between the current VR states and the values from the original VR it is only necessary to add the hard evidence to the merged state evolved from the one with evidence.
  Adding soft evidence $\boldsymbol{\epsilon}$ follows the next expression:
  $$p(V = \mathbf{V_{stm}} | \boldsymbol{\epsilon}) = \sum_{v \in \mathbf{V_{stm}}} p(v|\boldsymbol{\epsilon}) \quad (6)$$
- *Obtaining Inference:* In certain occasions an application could need the inference of an already merged (because not protected) state. Under the assumption that $p(v|\bigcup_{v_i \in \mathbf{V_{stm}}} v_i, \boldsymbol{\epsilon}) \approx p(v|\bigcup_{v_i \in \mathbf{V_{stm}}} v_i)$; $v \in \mathbf{V_{stm}}$, i.e. that the relation between the probabilities of the original states would stay unchanged with or without evidence, it can be calculated by:
  $$p(V = v|\boldsymbol{\epsilon}) \approx \frac{p(V = v)p(V = \mathbf{V_{stm}}|\boldsymbol{\epsilon})}{\sum_{v_i \in \mathbf{V_{stm}}} p(V = v_i)} \quad (7)$$
  Still this is only an approximation. For precise posterior calculation in that case, there is no alternative to switching back to the complete network.

E. Triggering

The following assumptions generally hold for ubiquitous, context-aware computing:

- In each moment the posterior probabilities are more related to the information given by the sensors than to their prior probabilities.
- Due to that and to the changing requirements of a service, the importance or the likelihood of each state changes faster than the conditional probabilities or the structure of the network.

Therefore we should provide methods to apply new repartitioning during system run-time to already partitioned networks, in order to extend the scope of the network beyond the restrictions of its simplified VR, to adapt it permanently to the newest needs.

To achieve this, different methods were investigated. An essential requirement for these is to maintain the original BN with complete VRs saved (but not necessarily in execution memory). One method developed consists in maintaining an *filtered probability* for each state of the current VR. We calculate it in the following way:

$$p_{filt}(v)_{t_n} = \alpha \cdot p_{filt}(v)_{t_{n-1}} + (1-\alpha) \cdot p(v|\boldsymbol{\epsilon}), \quad (8)$$

where $p_{filt}(v)_{t_n} = P_{filt}(V = v)_{t=t_{n-1}}$ is the filtered probability of the last inference process and $p(v|\boldsymbol{\epsilon})$ is the posterior probability for the state $V = v$ given evidence $\boldsymbol{\epsilon}$.

This filter preserves the characteristics of a probability distribution. Therefore it could be used as trigger for new repartitioning processes and in a state selection criterion (see III-A) instead of prior probabilities. As an example, a maximum accumulated probability $P_{ac}(V = v) \; \forall \; v = \bigcup_{i \in I} v_i \;\wedge\; \bar{I} > 1$ for all merged states can be calculated and saved. If this limit was exceeded by a new evaluation based on new evidence, a new repartitioning process would be started applying e.g. a minimum joint marginalization criterion based on the excess-probability.

Different repartitioning triggers are possible as well, e.g. a minimum Entropy for the VR.

IV. Evaluation

This section gives a first, simplistic evaluation of the described methodology.

A. Test description

We applied our approach to small scenarios like the following ones.

(a) Employees in a large campus:
Imagine a campus with several, say n buildings, each with tens of offices. It might make sense for some applications to infer location aware context of an employee only with information about their regular building, accumulating the rest of the campus as `Other`. Inference then has to cope with only $\frac{1}{n}^{th}$ of possible locations. If an employee changes building, also the value range of `location` should adapt to the new building.

(b) An activity dependent service:
In the entrance of a company there is a big wall display, showing relevant information depending on the activity (`Arriving`, `Leaving`, `GoingToLunch` or `Other`) of the employee passing it by. It is not using other activities, even if the context aware system could provide many more.

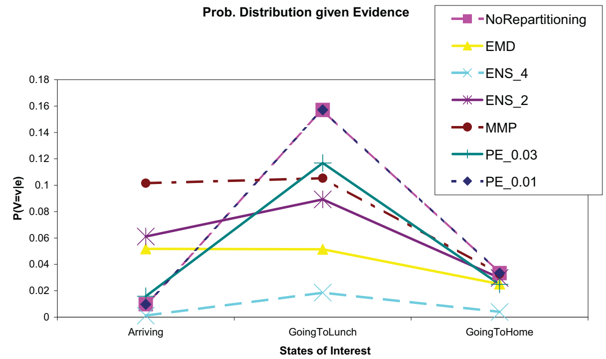

Figure 3. Probabilities for the states of interest of the node ‘Activity’, give certain evidence, for the original network and its repartitioned versions.

In these scenarios we selected a subset of interest according to III-B, e.g. the states `Arriving`, `GoingToLunch` and `GoingHome` of a node `activity` in scenario (b).

A *sample execution* of the program contained the following steps:

1) Network construction from files, i.e the construction of the necessary Java object hierarchy to allow for inference over the network.
2) Repartitioning
3) Inference
4) Bayesian Network storage

This process was executed for the repartitioning methods from the previous section and comparing it to a framework without dynamic value ranges. Inference was based on a PPTC implementation following Huang [10]. The compared repartitioning methods were ENS_N (where N indicates the number of states of each subset merged to obtain a new state), EMD in its binary form, MMP (merging of adjacent states), and PE_N (where N indicates the minimum level of mutual information between a state and its parent or child nodes of interest to be protected).

B. Costs for repartitioning

The quality of all methods must be measured taking into account eventual temporal wins, but also the loss in quality of information.

1) Error: To analyse the error introduced by the VR reduction, the posterior probabilities given by inference in the original BN were compared with the inference results from the repartitioned BNs. Inference was based on well-defined evidence introduced into the sensor nodes, in both original and repartitioned BNs.

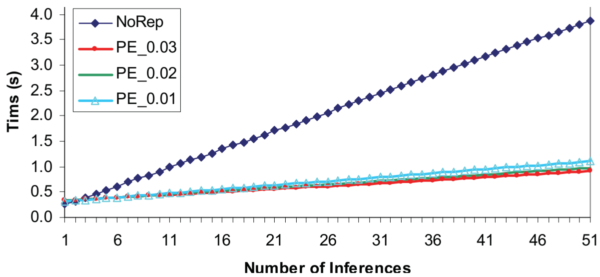

Figure 4. Development of inference time with the number of executions.

The plot of Figure 3 shows the results for the three service states of node `activity`. In this case, it is clear that the best performance for the states of interest is provided by the protection extension method that increases its quality by the reduction of the information limitation. However it implies a higher number of states than the remaining methods.

Analysing the relative error obtained for each state, we can observe, that the error of the inference with the PE methods is in overall lower than with other repartitioning approaches, but increases considerably for values out of the interest range.

2) Time: Our analysis confirmed that due to the complexity of a BN with value range processing capabilities, the BN construction time is increased slightly. Repartitioning itself proved to be much shorter than an inference process even for an small BN. The better accuracy of the PE methods implies an increased repartitioning time with respect to simpler methods like ENS. Taking into account that repartitioning only has to be realized once for many inference processes, this difference in CPU time is negligible for the overall system consumption. The wins in overall inference time depends on the number how often the repartitioned network is used for inferring. The corresponding curve is shown in Figure 4.

V. Conclusion

This work shows that dynamic modification of value ranges is feasible and entails a performance win. The preliminary evaluation from section IV has shown that the introduced error by cutting some information can be kept within reasonable limits and the repartitioning costs are outweighed by the reduced inference time. The more often a reduced BN is used, the higher is the efficiency win.

As a next step, we will apply this system in a complete context aware framework, connected among others to a UWB indoor positioning system and recognition of physical activity. From this integration and extensive usage, we expect realistic data in terms of update frequency and VR modification frequency, particularly for activity - one of the most important, but also one of the most challenging types of context information.

References

[1] J. Pearl, *Causality: Models, Reasoning, and Inference*. Cambridge University Press, 2000.

[2] G. F. Cooper, "Probabilistic inference using belief networks is NP-hard," Medical Computer Science Group, Knowledge Systems Laboratory, Stanford University, Stanford, CA, Tech. Rep. KSL-87-27, May 1990.

[3] U. Fayyad and K. Irani, "Multi-interval discretization of continuous-valued attributes for classification learning," in *Proceedings of the International Joint Conference on Uncertainty in AI*, 1993, pp. 1022–1027.

[4] R. Barco, P. Lázaro, L. Díez, and V. Wille, "Continuous versus discrete model in autodiagnosis systems for wireless networks." *IEEE Trans. Mob. Comput.*, no. 6, pp. 673–681, 2008.

[5] E. J. Clarke and B. A. Barton, "Entropy and mdl discretization of continuous variables for bayesian belief networks," *International Journal of Intelligent Systems*, vol. 15, no. 1, pp. 61–92, 2000.

[6] R. R. Bouckaert, "Properties of measures for bayesian belief network learning," in *In: Tenth Conference on Uncertainty in Artificial Intelligence*, 1994.

[7] N. Friedman and M. Goldszmidt, "Discretizing continuous attributes while learning Bayesian neworks," in *Proc. 13th International Conference on Machine Learning*. Morgan Kaufmann, 1996, pp. 157–165.

[8] A. V. Kozlov and D. Koller, "Nonuniform dynamic discretization in hybrid networks," in *In Proc. UAI*. Morgan Kaufmann, 1997, pp. 314–325.

[9] J. Dougherty, R. Kohavi, and M. Sahami, "Supervised and unsupervised discretization of continuous features," in *International Conference on Machine Learning*, 1995, pp. 194–202.

[10] C. Huang and A. Darwiche, "Inference in belief networks: A procedural guide," *International Journal of Approximate Reasoning*, vol. 15, no. 3, pp. 225–263, 1996. [Online]. Available: citeseer.nj.nec.com/huang94inference.html