GENERATIONS IN BAYESIAN NETWORKS

Alexander Litvinenko[1], Natalya Litvinenko[2], Orken Mamyrbayev[2], Assem Shayakhmetova[2]

[1]Chair of Mathematics for Uncertainty Quantification, RWTH Aachen, Aachen, Germany, [2]Institute of Information and Computational Technologies, Almaty, Kazakhstan

Abstract. This paper focuses on the study of some aspects of the theory of oriented graphs in Bayesian networks. In some papers on the theory of Bayesian networks, the concept of "Generation of vertices" denotes a certain set of vertices with many parents belonging to previous generations. Terminology for this concept, in our opinion, has not yet fully developed. The concept of "Generation" in some cases makes it easier to solve some problems in Bayesian networks and to build simpler algorithms.

In this paper we will consider the well-known example "Asia", described in many articles and books, as well as in the technical documentation for various toolboxes. For the construction of this example, we have used evaluation versions of AgenaRisk.

Keywords: Bayesian networks, AgenaRisk, oriented graphs, vertices generation

GENERACJE W SIECIACH BAYESOWSKICH

Streszczenie. Niniejszy artykuł koncentruje się na badaniu pewnych aspektów teorii zorientowanych grafów w sieciach bayesowskich. W niektórych artykułach na temat teorii sieci bayesowskich pojęcie „generacji wierzchołków" oznacza pewien zestaw wierzchołków z wieloma rodzicami należącymi do poprzednich generacji. Terminologia tego pojęcia, naszym zdaniem, nie została jeszcze w pełni rozwinięta. Koncepcja „Generacji" w niektórych przypadkach ułatwia rozwiązywanie niektórych problemów w sieciach bayesowskich i budowanie prostszych algorytmów.

W tym artykule rozważymy dobrze znany przykład „Azja", opisany w wielu artykułach i książkach, a także w dokumentacji technicznej różnych zestawów narzędzi. Do budowy tego przykładu wykorzystaliśmy wersje testowe AgenaRisk.

Słowa kluczowe: sieci bayessowskie, grafy zorientowane, generacja wierzchołków

Introduction

Nowadays artificial intelligence is widely used in different fields of science. Since the beginning of 2000 Bayesian Networks is the most popular tool of artificial intelligence in different researches. For the correct use of Bayesian Networks, the corresponding mathematical tools were developed. There are ample opportunities to study problems in various fields of science. However, massive, complex computations in the process of work with Bayesian networks required the involvement of both good computing equipment and the availability of good software that ensure convenient work with Bayesian networks.

The most popular software products for work with Bayesian Networks are BayesiaLab, AgenaRisk, Bayes Server, Netica, Hugin Expert, BayesFusion. Even though some packages are over 15 years old, there is still an issue with interaction between mathematicians working on the theoretical basis of Bayesian networks, algorithm developers and programmers, as well as researchers in applied science.

In this paper, we will consider one of the theoretical directions in the theory of Bayesian networks – partition of the vertices of the Bayesian network into generations. In our opinion, the theory of Bayesian networks is rather poorly represented in the literature; however, in some cases, partition vertices into generations may allow you to take a fresh look at the general theory of Bayesian networks, and make it easier to develop algorithms for some problems.

In this paper we will consider the well-known example "Asia", described in many articles and books, as well as in the technical documentation for various toolboxes. For the construction of this example, we have used evaluation versions of AgenaRisk. For the basic knowledge about Bayesian networks reader can use the literature [2, 4, 6–13] and the literature [1, 5] to get the basic knowledge about the capabilities of AgenaRisk.

1. Main definitions

Bayesian Networks are a handy tool for describing complex processes with uncertainties.

Below we list the basic concepts associated with Bayesian networks.

Definition 1. A Bayesian network is an acyclic oriented graph with Markov condition. The vertices of the graph are often called nodes. Nodes represent some variables that reflect the main entities in the developed model.

Definition 2. A Bayesian network is called discrete if the variables representing nodes are discrete.

Definition 3. A Bayesian network is called continuous if the variables representing the nodes are continuous.

Definition 4. A Bayesian network is called hybrid if there are nodes in the Bayesian network that represent both discrete and continuous variables.

Definition 5. A Bayesian network is called dynamic if the variables representing the nodes change over time.

Some nodes of the Bayesian network may be interconnected by directed arcs (edges). Arcs define some probabilistic connection between corresponding nodes. Sometimes such a relationship is causal. The reason is the node where the oriented arc comes from, the consequence is the node where the oriented arc comes

Definition 6. The edges are arcs without direction.

Definition 7. If the nodes are not connected by an arc, then these nodes are considering as conditionally independent.

Definition 8. A skeleton of a Bayesian network is a graph obtained from a Bayesian network by replacing arcs with edges.

Definition 9. If an arc goes from the vertex A to the vertex B, then A is called the parent of B, and B is called the child vertex of the vertex A.

Definition 10. Let Y be some subset of vertices of a Bayesian network. P (Y) is often denoted as the set of all parents belongs to Y. C (Y) is often denoted as the set of all children belongs to Y.

Definition 11. A node x and an arc e are called incident if e enters or comes from x.

Definition 12. Nodes are called supplementary if they are incidents of one arc, i.e. if one of them is the parent of the other one.

Definition 13. The route from vertex a_0 to vertex a_n in an oriented graph (in a Bayesian network) is an alternating sequence of vertices and arcs of the form a_0, $\{a_0\ a_1\}$, a_1, $\{a_1\ a_2\}$, a_2, $\{a_2\ a_2\}$, ⋅⋅⋅, a_n.

Definition 14. The path is a route without repeating arcs.

Definition 15. If there is an oriented path from the vertex A to the vertex B, then A is called the ancestor of B, and B is called the descendant of A.

Definition 16. Two nodes are called connected if there is a route between them.

Definition 17. If a vertex has no ancestors, then its local probability distribution is called unconditional, otherwise conditional.

Definition 18. The topological numbering of nodes of a Bayesian network is called such a numbering of nodes, in which the number of any node is greater than its parent number.

Definition 19. Sometimes it is convenient to part nodes into generations to develop Bayesian network calculation algorithms. Generations can be of two types: generations of descendants and generations of ancestors. Generations of descendants are defined as follows:

- Nodes without parents belong to the 0 generation of descendants.
- Nodes with only 0 generation of parents belong to 1 generation of descendants.
- Nodes with only 0 and 1 generation of parents belong to 2 generation of descendants.
- ………….
- Nodes with 0, 1, 2, … K generation of parents belong to K+1 generation of descendants
- ………….

Definition 20. Generations of ancestors are defined as follows:

- Nodes with no children belong to the 0 generation of ancestors.
- Nodes with only 0 generation of children belong to 1 generation of ancestors.
- Nodes with only 0 and 1 generation of children belong to the 2 generation of ancestors.
- ………….
- Nodes with only **0, 1, 2, … K** generation of children belong to the **K+1** generation of ancestors.
- …………..

2. How can we use the definition "Generation"?

We can distinguish two types of generations – generations of descendants and generations of ancestors. Generations of descendants are built, starting from vertices without parents. Generations of descendants are built, starting from vertices without children. In most cases, we obtain completely different partitions that have a different number of generations. However, it is easy to build a graph (Bayesian network) which partition into generations of descendants and generations of ancestors are completely identical.

The expediency of partition vertices of a Bayesian network into one or another type of generation is determined by a specific task. To solve some problems, sometimes we need to do two partitions at once: into generations of descendants and generations of ancestors.

The algorithm for partition vertices of a Bayesian network into generations is quite simple. Let's consider the both variants.

Partition into generations of descendants

First, we search and select vertices without parents. We assign such vertices to zero generation and mark selected vertices. Next, we look through the remaining unmarked vertices and select only those vertices whose parents belong only to the zero generation. We obtain the first generation of descendants, and mark the newly selected vertices of the first generation. Again, we look through the remaining unmarked vertices and select only those vertices whose parents belong only to either zero or first generation. We obtain the second generation of descendants and mark the selected vertices. We continue that way until all vertices are marked.

An example of the partition into generations is shown in Figure 1.

Generations of descendants for this Bayesian network:

- Vertices **Age** and **VisitAsia** will be referred to the zero generation.
- Vertices **Smoker** and **Tuberculosis** belong to the first generation.
- Vertices **Cancer** and **Bronchtis** belong to the second generation.
- The only vertex **TbOrCa** belong to the third generation.
- Vertices **XRay** and **Dyspnea** belong to the fourth generation.

Partition into generations of ancestors

First, we search and select vertices without children. We assign such vertices to zero generation of ancestors and mark selected vertices. Next, we look through the remaining unmarked vertices and select only those vertices whose children belong only to the zero generation of ancestors. We obtain the first generation of ancestors, and mark the newly selected vertices of the first generation. Again, we look through the remaining unmarked vertices and select only those vertices whose children belong only to either zero or first generation of ancestors. We obtain the second generation of descendants and mark the selected vertices.
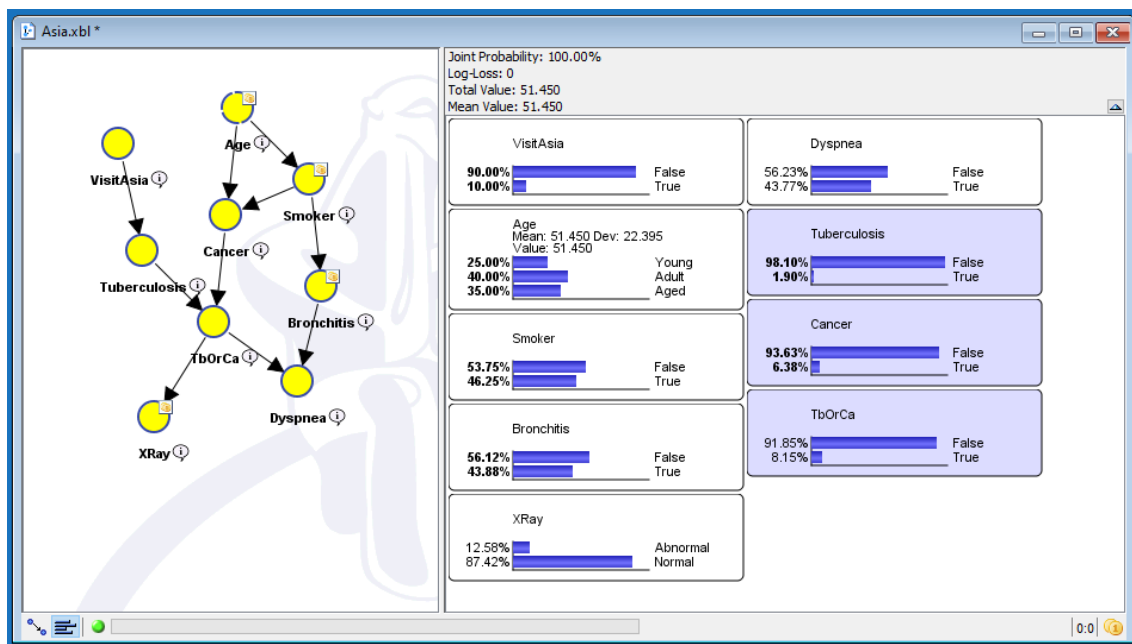

Fig. 1. Example of the partition intogenerations

We continue that way until all vertices are marked.

Generations of ancestors for this Bayesian network:

- Vertices **XRay** and **Dyspnea** will be referred to the zero generation.
- Vertices **TbOrCa** and **Bronchtis** belong to the first generation.
- Vertices **Cancer** and **Tuberculosis** belong to the second generation.
- Vertices **Smoker** and **VisitAsia** belong to the third generation.
- The only vertex **Age** belong to the fourth generation.

We can use both partitions into generations: into generations of descendants or generations of ancestors to prove the following theorem.

Theorem 1. *If at some stage of generation's construction, some generation is empty, but we still have unmarked vertices, there are oriented cycles in the graph.*

Proof.

If at some stage among unmarked vertices there were no vertices that have parents from previous generations, then considered node can only have as parents the nodes from the current generation. We choose any parent whose parents also do not belong to previous generations. As a result, we have an unlimited path with a limited number of nodes, that is, a cycle.

Thus, we have a fairly simple way to find out if an oriented graph has a cycle and find where this cycle is.

Below are a few standard, previously proven, theorems from graph theory and the theory of Bayesian networks, as well as their new proofs with using the concept of "Generation". In our opinion, the proofs of these theorems are simpler.

Theorem 2. *Two isomorphic Bayesian networks have the same partition into generations of descendants and generations of ancestors.*

Proof.

1. If the partition of the vertices of the two Bayesian networks into generations of descendants have done, and the partitioning data for these networks are different, then in the generation where there is a difference, there are vertices which have a different number of parents in each network. This fact contradicts the isomorphism condition of two networks.
2. If the partition of the vertices of the two Bayesian networks into generations of ancestors have done, and the partitioning data for these networks are different, then in the generation where there is a difference, there are vertices which have a different number of children in each network. This fact contradicts the isomorphism condition of two networks.

The problem of verifying the isomorphism of two Bayesian networks is greatly simplified by using the concept of "Generation".

Theorem 3. *Bayesian network allows topological numbering.*

Proof.

Let the vertices of the Bayesian network be partitioned into generations of descendants.

The first method of numbering

First, we randomly number the nodes of the zero generation, then continue the numbering for the nodes of the first generation, then for the nodes of the second generation, etc.

Since the parents of each node are in previous generations, the numbering will be topological.

The second method of numbering

Let's consider all nodes of zero generation. We take an arbitrary node of the zero generation and consider the set **M1** of all descendants of this node on all generations. We number the nodes of the set **M1** as follows. We assign the number 1 to the node from the zero generation. Next, we number the nodes of the first generation on set **M1**, then the nodes of the second generation on set **M1**, etc. Then we mark all the numbered nodes.

Next, we take the next node of the zero generation and assign the next number to it. We consider the set M2 of all descendants of a given node which was not previously marked. Next, we number the nodes of the first generation on set **M2**, then the nodes of the second generation on set **M2**, etc. Then we mark all the numbered nodes.

We do the same with all the remaining nodes of zero generation. We number these nodes in the same way.

Since the parents of each node are from previous generations, the numbering will be topological.

Theorem 4. *The Bayesian network allows such a topological numbering, in which the number of any node X will be greater than the numbers of any node from an arbitrary set of nodes M that do not contain node X, as well as descendants of node X.*

Proof.

Let the vertices of the Bayesian network be partitioned into generations of descendants.

Let's slightly change the second numbering method from Theorem 3.

Let **MX** be the set of nodes of the considered Bayesian network, consisting of the node **X** and its descendants. Let **L** be the number of nodes in the considered Bayesian network. We will do the topological numbering of network nodes by the second method described in Theorem 3.

We add the number **L** to all numbers of nodes of the set **MX**. Continuous numbering of Bayesian network nodes will be broken. However, the number of any node will remain still greater than the number of its parent. Now we need to restore the numbering continuity of the Bayesian network.

We arrange in ascending order the obtained node numbers of the Bayesian network. Let's number these node numbers. If we take the numbering of node numbers for new node numbers, then this numbering will remain topological numbering. In addition, the node numbers of the **MX** array will be greater than the numbers of all remaining nodes of the network, and therefore greater than node numbers of the array **M**.

Theorem 5. *The Bayesian network allows such a topological numbering, in which the number of any node X will be less than the numbers of any node from an arbitrary set of nodes M that do not contain node X, as well as ancestors of node X.*

Proof.

Let the vertices of the Bayesian network be partitioned into generations of ancestors.

Let's slightly change the second numbering method from Theorem 3.

Let **MX** be the set of nodes of the considered Bayesian network, consisting of the node **X** and its ancestors. Let **L** be the number of nodes in the considered Bayesian network. We will do the topological numbering of network nodes by the second method described in Theorem 3.

We add the number **L** to all numbers of nodes of the set **MX**. Continuous numbering of Bayesian network nodes will be broken. However, the number of any node will remain still less than the number of its child. Now we need to restore the numbering continuity of the Bayesian network.

We arrange in ascending order the obtained node numbers of the Bayesian network. Let's number these node numbers. If we take the numbering of node numbers for new node numbers, then this numbering will remain topological numbering. In addition, the node numbers of the **MX** array will be less than the numbers of all remaining nodes of the network, and therefore less than node numbers of the array **M**.

3. Conclusion

This paper shows some possibilities of using the concept of "Generation" in the process of proof of some standard theorems in the theory of Bayesian networks. Use of this concept also simplifies some algorithms for problems in Bayesian networks, for example, searching for loops or checking the isomorphism of two networks.

Here we have presented only a small part of the possibilities of using the concept of "Generation".

Acknowledgements

References

[1] AgenaRisk 7.0 User Manual. 2016.
[2] Bidyuk P., Terentyev A.: Construction and methods of learning of Bayesian Networks. Tavricheskiy vestnik informatiki i matematiki 2/2004, 139–154.
[3] Getting Started with AgenaRisk. 2013.
[4] http://web4.cs.ucl.ac.uk/staff/D.Barber/textbook/020217.pdf (available 15.05.2019).
[5] http://www.agenarisk.com (available 15.05.2019).
[6] http://www.businessdataanalytics.ru/download/BayesianNetworks.pdf (available 15.05.2019).
[7] http://www.cs.technion.ac.il/~dang/books/Learning%20Bayesian%20Networks(Neapolitan,%20Richard).pdf (available 15.05.2019).
[8] http://www.stat.yale.edu/~jtc5/BioinformaticsCourse2001/MurphyBayesNetIntro.pdf (available 15.05.2019).
[9] https://pdfs.semanticscholar.org/7bc7/54bc548f32b9ac53df67e3171e8e4df66d15.pdf (available 15.05.2019).
[10] Jensen F. V., Nielsen T. D.: Bayesian Networks and Decision Graphs. Springer, 2007.
[11] Litvinenko N., Litvinenko A., Mamyrbayev O., Shayakhmetova A.: On the issue of classification of types of evidence in Bayesian networks. IPIC, Almaty 2018.
[12] Litvinenko N., Litvinenko A., Mamyrbayev O., Shayakhmetova A.: Work with Bayesian Networks in BAYESIALAB. IPIC, Almaty 2018.
[13] Murphy K. P.: Machine Learning A Probabilistic Perspective. MIT Press, 2012.

Ph.D. Alexander Litvinenko
e-mail: litvinenko@uq.rwth-aachen.de

Alexander Litvinenko joined RWTH Aachen in 2018. He specializes in efficient numerical methods for stochastic PDEs, uncertainty quantification, multilinear algebra and Bayesian update methods. Alexander earned B.Sc. (2000) and M.Sc. (2002) degrees in mathematics at Novosibirsk State University, and his Ph.D. (2006) at Max-Planck Institute in Leipzig. From 2007-2013 he was a Postdoctoral Research Fellow at the TU Braunschweig in Germany. From 2013-2018 he worked as Research Scientist in KAUST, Saudi Arabia.

ORCID ID: 0000-0001-5427-3598

M.Sc. Natalya Litvinenko
e-mail: n.litvinenko@inbox.ru

Natalya Litvinenko earned B.Sc. (2012) and M.Sc. (2014) in Mathematics at the Kazakh National University named after Al-Farabi, she also did an internship at Imperial College London (2014). Her research interest includes implementation of parallel algorithms on architectures NVIDIA CUDA, and Bayesian Networks. From 2018 till now she is taking part in the project "Development and software implementation of a package for solving applied problems in Bayesian networks".

ORCID ID: 0000-0002-0576-8305

Ph.D. Orken Mamyrbayev
e-mail: morkenj@mail.ru

Orken Mamyrbayev earned B.Sc. (2001) and M.Sc. (2004) in Information systems at the Kazakh National Research Technical University named after K.I.Satpayev. His research interest includes digital signal processing, robotic systems, computer vision, automatic speech recognition. He earned his Ph.D. (2014) in the group of Prof. Maksat Kalimoldayev. Currently he is deputy director in the Institute of Information and Computational Technologies CS MES RK.

ORCID ID: 0000-0001-8318-3794

Ph.D. Assem Shayakhmetova
e-mail: a_sh_s83_83@mail.ru

Assem Shayakhmetova earned B.Sc. (2004) and M.Sc. (2006) in Computer science. She earned Ph.D. (2017) in Information systems at the Kazakh National Research Technical University named after K.I.Satpayev. She also did an internship at Imperial College London (2015). Her research interest includes the mathematics, information science and control theory.

ORCID ID: 0000-0002-4072-3671