Enhancements to the Bayesian Network for Raster Data (BayNeRD)

Alexsandro C. de O. Silva[1], Marcio P. Mello[2], Leila M. G. Fonseca[1]

[1]National Institute for Space Research (INPE)
Image Processing Division (DPI)
Avenida dos Astronautas 1758 – 12201-010 – São José dos Campos, SP – Brazil

[2]The Boeing Company
Boeing Research & Technology – Brazil (BR&TB)
Estrada Dr Altino Bondesan 500 – 12247-016 – São José dos Campos, SP – Brazil

{acos,leila}@dpi.inpe.br, marcio.p.mello@boeing.com

Abstract. *Bayesian Networks are powerful probabilistic method to make inferences based on evidences. However, this technique has been rarely applied to processing Remote Sensing data. The Bayesian Network for Raster Data – BayNeRD, implemented in the R software, was developed to be used for raster data analysis. This paper describes an enhanced version of the BayNeRD algorithm, stating what has been changed in terms of data pre-processing, user interaction, and outputs.*

1. Introduction

A system that can work with information absence (uncertainty) should be able to assign reliability levels to the variables in its knowledge base and set relations among them. Bayesian Networks (BNs) are probabilistic methods that use graph theory to set relations between variables and probability theory to assign belief levels [Marques and Dutra 2008; Qin et al. 2006].

According to Pearl and Russel (2000), the ability for bidirectional inference, combined with a rigorous probabilistic base, led to the emergence of BNs as a method for reasoning in artificial intelligence and expert system. Indeed, BNs have been used with focus on several applications, such as: medical diagnosis [Kahn et al. 1997; Oniśko and Druzdzel 2013]; speech and gesture recognition [Daoudi et al. 2003; Suk et al. 2010]; and Environmental Science [Castelletti and Soncini-Sessa 2007; Dlamini 2010; Gambelli and Bruschi 2010]. However, even though BN applications have been increasing over the last two decades, Aguilera et al., (2011) and Qin et al. (2006) pointed out that BNs has been rarely used in Earth Observation Science and to process Remote Sensing data.

In this context, Mello et al. (2013) developed a computer-aided Bayesian Network method able to incorporate expert knowledge for the benefit of remote sensing applications and other raster data analysis: Bayesian Network for Raster Data (BayNeRD). It was implemented in R software [R Core Team 2014] and a case study of soybean mapping in Mato Grosso State, Brazil, was used to evaluate its capability to model complex phenomena through plausible reasoning based on data observation. Although the main concepts of BNs for remote sensing applications were implemented by Mello et al. (2013), there are many improvements that can be implemented in order to create an enhanced version of the BayNeRD algorithm. Thus, this paper describes the main enhancements implemented in the BayNeRD algorithm to improve it as well as the changes in terms of data pre-processing, user interaction, and program outputs.

2. Bayesian Networks and Inference Background

BNs, sometimes called Bayesian Belief Networks [Uusitalo 2007], are mathematical models based on two components: (i) a Directed Acyclic Graph (DAG), and (ii) conditional probability tables (CPTs) [Landuyt et al. 2013].

In a DAG, each node represents a variable in the model, while an arrow linking two variables indicates dependence between them. An arrow starts in a parent variable and ends in a descendent one. As the graph is acyclic, there is no feedback arrows from a descendent to a parent [Landuyt et al. 2013]. One advantage of BNs over other types of predictive models, such as decision trees and neural networks, is that unlike those "black box" approaches, the DAG's arrows represent real connections, not flow of information [Pearl & Russell 2000; Qin et al. 2006]. Figure 1 shows an example of a BN, in which the variable X_1 influences X_2 and X_3. Furthermore, X_3 is influenced by X_2.

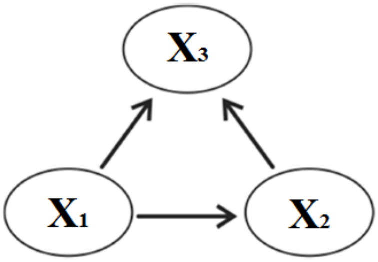

Figure 1. Example of Bayesian Network graphical model.

Once the graphical model is defined, it is necessary to know the probability relations. If a node has no parent, a prior probability function is assigned to it: $P(X_i = x_i)$, in which $x_1, \ldots, x_n$ are all possible values (i.e., instantiations) of variable X_i. On the other hand, if a node has parent(s), a conditional probability function is assigned to it. For each variable, the function expresses the probability of that variable being in a particular state (instantiated), given the states of its parents. If x_i denotes the values of the variable X_i and pa_i denotes the set of values for X_i's parents, then $P(x_i|pa_i)$ denotes the conditional probability. For example (see Figure 1), $P(X_3 = x_3|X_1 = x_1, X_2 = x_2)$.

The fundamental rule of conditional probability is

$$P(a|b) = P(a,b)/P(b) \qquad (1)$$

or

$$P(a|b) * P(b) = P(a,b) \qquad (2)$$

in which $P(a,b)$ is the joint probability of event $a \wedge b$.

It is known that $P(a,b) = P(b,a)$. Thus, from Eq. 2 we have

$$P(a|b) * P(b) = P(b|a) * P(a) \qquad (3)$$

that will result in

$$P(a|b) = P(b|a) * P(a)/P(b). \quad (4)$$

Eq. 4 is the Bayes' theorem [Neapolitan 2003], which is the core of a BN, allowing probabilities to be updated under the light of a new evidence. Indeed, the Bayes' theorem updates the knowledge (priori probability) of an event considering new/additional evidence (conditional probability), allowing one to have a plausible reasoning based on a degree of belief (posteriori probability). This ability to compute posteriori probabilities given some evidence is called inference.

The effectiveness of a BN lies in the possibility to compute through Bayes' theorem not only the probability distributions for descendent variables given the values of their parents, but also the distributions of the parents given the values of their descendants. That is, the BN allows to know the effects given the causes and the causes given the effects [Uusitalo 2007; Aguilera et al. 2011].

The joint distribution is computed by the product of prior and conditional probabilities for each variable given its parents, as

$$P(x_1, \dots, x_n) = \prod_{i=1}^{n} P(x_i|pa_i) \quad (5)$$

in which x_i is the value observed for the variable X_i and pa_i is the set of values for X_i's parents.

3. The original BayNeRD algorithm

The first version of the BayNeRD algorithm implemented by Mello et al. (2013) is available on Internet. Figure 2 shows the procedures used to apply the BayNeRD system in a study case to identify soybean plantations in Mato Grosso State, Brazil.

The BayNeRD algorithm has been implemented in R software [R Core Team 2014]. The algorithm handles raster data in GeoTiff format, where each GeoTiff corresponds to a variable (node) of the network. The variable that represents the phenomenon under study is named *target variable* and the remaining variables are named *context variables*.

There are packages already implemented in R software to handle spatial data. GeoTiff files were originally loaded by *rgdal package* [Bivand et al. 2014], which makes bindings to Geospatial Data Abstraction Library (GDAL).

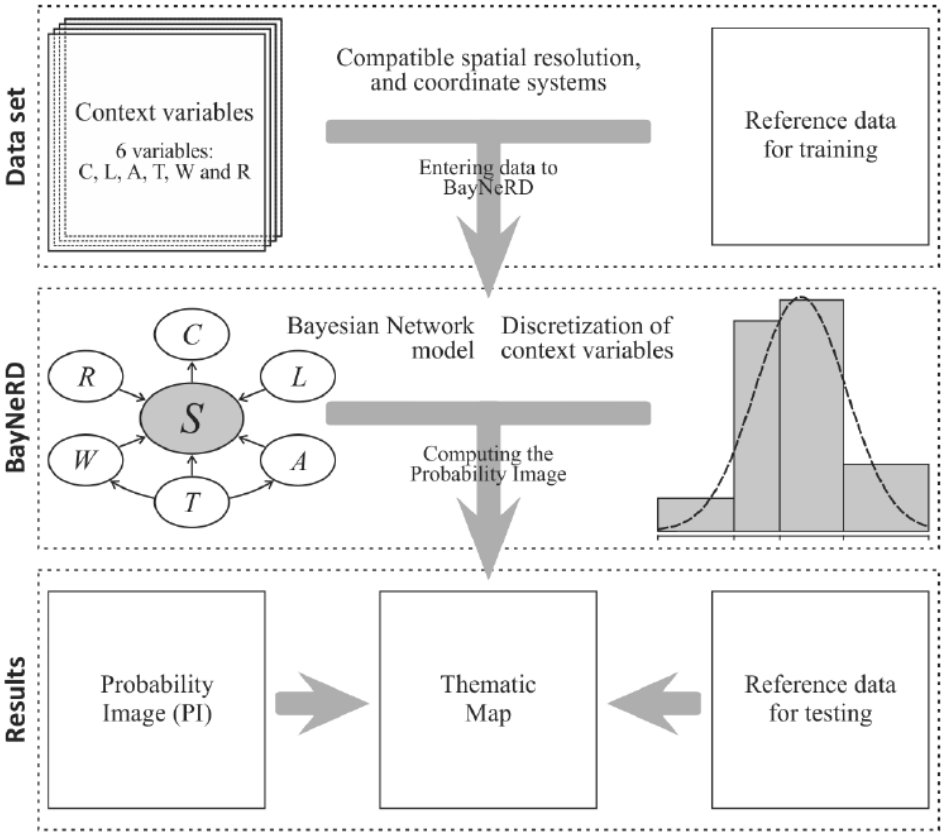
Data set
Context variables
6 variables:
C, L, A, T, W and R
Compatible spatial resolution, and coordinate systems
Entering data to BayNeRD
Reference data for training
BayNeRD
C
R
L
S
W
A
T
Bayesian Network model
Discretization of context variables
Computing the Probability Image
Results
Probability Image (PI)
Thematic Map
Reference data for testing

Figure 2. Procedures for using BayNeRD in remote sensing applications [Mello et al. 2013].

3.1. Target and Context Variables

The GeoTiff that represents the target variable must be provided as reference data for training. Its pixels can be categorized in four labels: (i) target presence; (ii) target absence; (iii) missing data (no observation); and (iv) pixels outside the study area. Although the target variable GeoTiff may contain more than four labels, it must have at least two: (i) and (ii) [Mello et al. 2013].

The context variables can be either numerical (e.g., pixels with digital numbers) or categorical (e.g., thematic map); and they can contain also missing data. The context variables can have dependence relationship with target variable and/or among themselves. Furthermore, in the original version of BayNeRD, all GeoTiffs (for the target variable as well as for all context ones) must be compatible in relation to pixel size and coordinate system and all variables (target and context) must represent the same geographic area.

3.2. Bayesian Network Graphical model

After reading the target and the context variables into BayNeRD, the algorithm interacts with the user to design a BN graphical model. In the original version, the user is asked about the dependence relations among all variables. As dependencies are represented by arrows in a DAG, BayNeRD asked if there was an arc from a variable to all the others (pairwise).

Among the packages already implemented in R software to compute Bayesian analysis, the *catnet package* [Balov and Salzman 2013] has been developed to handle discrete BNs.

3.3. Discretization and Probability Functions

One of the main difficulties of applying BNs for real problems is to define the probability functions. According to Mello et al. (2013), BayNeRD was developed to interact with the user to define the probability functions through discretization processes of the context variables based on his/her knowledge. In the discretization, observed values of a variable are represented by discrete quantities (similar to process of drawing a histogram). In other words, the range of observed values is divided into intervals defined by the user.

In BayNeRD, the number of intervals can be computed following three criteria: (i) equidistant, in which intervals with the same width, (ii) quantile, in which intervals tends to have the same number of elements (pixels), and (iii) manually, in which the user defines the limits of each interval. The discretization will impact the probability functions, which are computed through pixel counting to both the dependence relations defined in the graphical model and the intervals defined in the discretization process. The user should be skilled to define the suitable intervals for each context variable so that all scenarios (i.e., combination of variable's and its parent's intervals) have representative data to compute the probability functions [Mello et al. 2010]. Besides, defined probability functions shall be representative of the real probability functions.

3.4.Probability Image

When the probability of target occurrence is computed for each pixel inside the study area, given a specific scenario, a Probability Image (PI) is formed. The PI consists of a raster data and it is the main result of BayNeRD.

Suppose that in the example of Figure 1 the variable X_3 is the target variable. The probability image will be formed by computing

$$P(X_3 = 1 | X_1 = x_1, X_2 = x_2) \tag{7}$$

for each pixel in the study area; where 1 is used to represent target presence.

4. BayNeRD enhanced version

To enhance the BayNeRD algorithm and to keep it updated with new R packages, several improvements have been added to it. Following, we will describe the changes implemented in BayNeRD through a comparative process between the original and the enhanced version.

As mentioned before, there are some packages that handle raster data in the R software [R Core Team 2014] (such as *rgdal package*, used in the BayNeRD original version). The *'raster' package* [Hijmans et al. 2014] provides several high-level raster data manipulation functions and showed to be more suitable for BayNeRD. A noteworthy feature of this package is that it allows working better with large raster datasets stored on disk, which makes the processing faster, since the package does not read all the values of cell's raster into memory. The package creates objects from large files that only contain information about the data (e.g., filename, pixel size, extent) and, in computations with the pixels, the data are processed in chunks [Hijmans 2014].

Considering the advantages of *'raster' package* over the *rgdal package*, we added it into BayNeRD in order to optimize the raster data (GeoTiff) processing for both target and context variables.

4.1. Target variable

In the original BayNeRD version, the target variable is instantiated in two values: 1 and 0, representing presence and absence, respectively. In the enhanced BayNeRD version, the target variable may have more than two instantiations. For instance, the target variable raster may have label 1 for the thematic class A, label 2 for the thematic class B and 0 for the target absence. This change will affect the definition of probability functions and BayNeRD's outputs. In addition, the raster data may still contain labels representing missing data and also labels representing pixels outside the study area as before, which will be used only to mask out pixels outside the study area in the context variables.

4.2. Context variables

One of the main benefits of enhancing BayNeRD with the *'raster' package* is to reduce the preprocessing data step. In the enhanced BayNeRD version, raster data may have different structures unlike the original version, in which all raster data must have compatible pixel size, extent (number of rows and columns) and coordinate reference system.

Each input context variable in the BayNeRD system is transformed to be compatible with target variable. That is, through *'raster' package* functions the algorithm is able to transform the coordinate reference system, to resample the spatial resolution, and to intersect the raster data according to the bounding box of target variable as shown in Figure 3. In this case, a new raster data is created following the target variable's bounding box. The pixels are interpolated from the context variable using the nearest neighbor method to minimize distortions. If a region has no intersection between the context and the target variables, their pixels are filled with NA values (Not Available), indicating missing value.

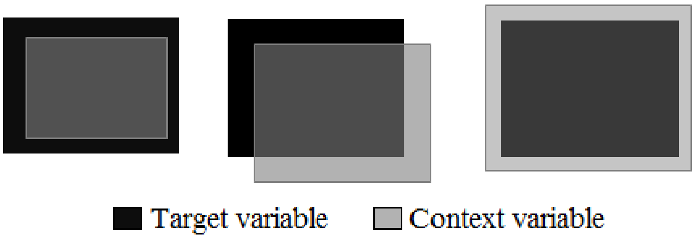

Figure 3. Different possibilities for intersection between context variable and target variable.

4.3. Bayesian Network Graphical model

The interactive process to state dependence relationships among the variables was substantially enhanced in BayNeRD. Before improvements, R interacted with the user by command line asking whether an arc exist for each pairwise variable. That is, for each context variable the algorithm ask $(n-1)$ questions, in which n is the number of variables. Therefore, the user must answer $n*(n-1)$ questions, that is, the growth of configuration questions is exponential.

In the enhanced BayNeRD, we implemented a more efficient way to define dependence relationships among the variables. The R software has also packages to handle with DAGs, and to perform Bayesian analysis. We created a new BN graphical model interface through the integration of both packages *'deal'* [Bottcher and Dethlefsen 2013] and *'bnlearn'* [Scutari 2009]. The first one allows the user to specify a BN through a point and click interface and the second one is used for Bayesian analysis and inference. Figure 5 shows the new interface implemented in the enhanced BayNeRD version.

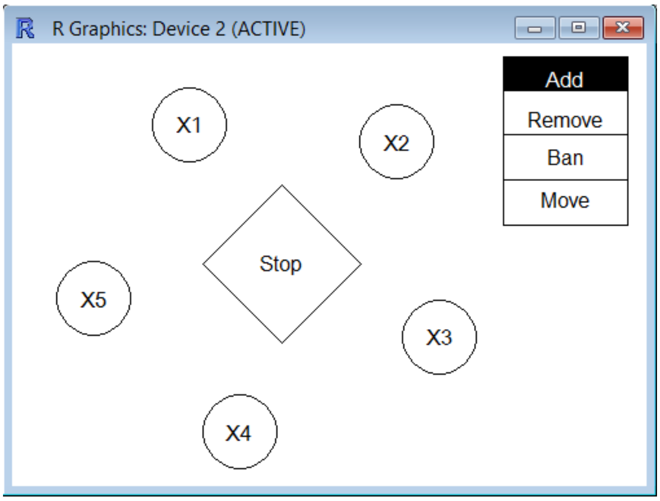

Figure 4. Interface to design the Bayesian Network graphical model.

In this interface, the user can insert or remove arcs between pairwise variables using either 'Add' or 'Remove' buttons and clicking on the variables. To insert/remove an arrow from node X_1 to node X_2, first click on "Add"/"Remove" button, then click on node X_1 and then click on node X_2. The interface also helps the user to avoid mistakes. For example, if the user enters an arrow to create a cycle it will not be drawn since the BN graphical model shall represent a DAG (acyclic). Therefore, the task of BN graphical model building becomes now easier and quicker compared to the previous BayNeRD version.

4.4. Discretization and Probability Functions

During the discretization process, the intervals can be defined by the user or following the criteria mentioned earlier – equidistance and quantile. These criteria were kept in the enhanced BayNeRD version and additional criterion was added. Now, the user may choose clustering criteria to convert continuous variables into categorical variables. This criteria is an unsupervised method through k-means clustering to partition the observed values of each context variables into k clusters defined by user, in which each value belongs to the cluster with the nearest mean [Hahsler et al. 2014].

The range of observed values for a context variable are divided into intervals, defined by lower and upper limits, and each interval is encoded as one category. Thus, a context variable will have n categories if it is discretized into n intervals. The probability functions are computed by counting categorical pixels for each context variable and taking into account their (in)dependence relationships. However, the user must also perform the discretization process for the target variable once the target variable may contain more than two labels in the enhanced BayNeRD version.

4.5. Probability Image

The number of instantiated values for the target variable defines also the number of output layers in the Probability Image (PI). Let's suppose that the variable X_3 (in Figure 1) is the target variable and the phenomenon occurrence is represented by three thematic classes: $X_3 = 1$, $X_3 = 2$ or $X_3 = 3$. In this case, we have one output layer with probability values for each class (presence). Each output layer is now called Probability Band (PB), and the stacking of PBs creates the Probability Image (PI) in the enhanced BayNeRD version.

Through the *bnlearn package*, queries are performed into CPTs and the system returns the probability of an event, given all observed evidences. A combination of evidences stated by the instantiation of all observed context variable is called scenario. Given the example in Figure 1, the PBs are computed, respectively, as

$$P(X_3 = 1|X_1 = x_1, X_2 = x_2) \tag{8}$$

$$P(X_3 = 2|X_1 = x_1, X_2 = x_2) \tag{9}$$

$$P(X_3 = 3|X_1 = x_1, X_2 = x_2). \tag{10}$$

For a given scenario, there are three events to compute associated probabilities. Consequently, the PBs are computed simultaneously to optimize the algorithm. If any context variable has missing data for any specific pixel, the probability for this pixel is computed anyway. It is even possible to compute $P(X_3 = x_3)$ for pixels without any observation through the priori probability, as in the original version.

5. Conclusion

The Bayesian Network for Raster Data – BayNeRD proposed by [Mello et al., 2013] represents a new probabilistic approach for raster data applications. This paper described the improvements made upon the original BayNerD algorithm that provided an enhanced BayNeRD version. Although this version has not been extensively tested, we do believe that it is more efficient and more user-friendly.

BayNeRD was enhanced to best handle raster data, also minimizing preprocessing of raster images. Indeed, the enhanced BayNeRD version handles raster datasets with different coordinate systems, spatial resolution and extent. A new BN graphical model interface is the most noticeable improvement in BayNeRD. It was also added the ability to model more than two possible instantiations for the target variable. The algorithm computes the probability for each possible target value given the observations made upon the context variables and then creates several Probability Bands (PB) that, when stacked, produce a Probability Image (PI) as outcome.

As future research, we plan to include spatial features (neighborhood information, for example) in the probability computation, and also to improve the algorithm efficiency by using parallel processing techniques. Also, we plan to exhaustively test the enhanced version comparing to the original one. The enhanced

BayNeRD version is still under construction and it will be available on Internet as soon as it is ready.

Acknowledgements

The authors thank CNPq (134400/2013-5) for financial support to first author; and the reviewers for their valuable comments and inputs.

References

Aguilera, P. a., Fernández, a., Fernández, R., Rumí, R., & Salmerón, a. (2011). Bayesian networks in environmental modelling. *Environmental Modelling & Software*, *26*(12), 1376–1388. doi:10.1016/j.envsoft.2011.06.004

Balov, N., & Salzman, P. (2013). catnet: Categorical Bayesian Network Inference. Retrieved from http://cran.r-project.org/package=catnet

Bivand, R., Keitt, T., & Rowlingson, B. (2014). rgdal: Bindings for the Geospatial Data Abstraction Library. Retrieved from http://cran.r-project.org/package=rgdal

Bottcher, S. G., & Dethlefsen., C. (2013). deal: Learning Bayesian Networks with Mixed Variables. Retrieved from http://cran.r-project.org/package=deal

Castelletti, a., & Soncini-Sessa, R. (2007). Bayesian Networks and participatory modelling in water resource management. *Environmental Modelling & Software*, *22*(8), 1075–1088. doi:10.1016/j.envsoft.2006.06.003

Daoudi, K., Fohr, D., & Antoine, C. (2003). Dynamic Bayesian networks for multi-band automatic speech recognition. *Computer Speech & Language*, *17*(2-3), 263–285. doi:10.1016/S0885-2308(03)00011-1

Dlamini, W. M. (2010). A Bayesian belief network analysis of factors influencing wildfire occurrence in Swaziland. *Environmental Modelling & Software*, *25*(2), 199–208. doi:10.1016/j.envsoft.2009.08.002

Gambelli, D., & Bruschi, V. (2010). A Bayesian network to predict the probability of organic farms' exit from the sector: A case study from Marche, Italy. *Computers and Electronics in Agriculture*, *71*(1), 22–31. doi:10.1016/j.compag.2009.11.004

Hahsler, M., Buchta, C., Gruen, B., & Hornik, K. (2014). arules: Mining Association Rules and Frequent Item sets. Retrieved from http://cran.r-project.org/package=arules

Hijmans, R. J. (2014). Introduction to the "raster" package (version 2.2-31). Retrieved from http://cran.r-project.org/web/packages/raster/vignettes/Raster.pdf

Hijmans, R. J., Etten, J. van, Mattiuzzi, M., Sumner, M., Greenberg, J. A., Lamigueiro, O. P., … Shortridge, A. (2014). raster: Geographic data analysis and modeling. *R Package Version 2.2-12*. Retrieved from http://cran.r-project.org/web/packages/raster/

Kahn, C. E., Roberts, L. M., Shaffer, K. A., & Haddawy, P. (1997). Construction of a Bayesian network for mammographic diagnosis of breast cancer. *Computers in Biology and Medicine*, *27*(1), 19–29. doi:10.1016/S0010-4825(96)00039-X

Landuyt, D., Broekx, S., D'hondt, R., Engelen, G., Aertsens, J., & Goethals, P. L. M. (2013). A review of Bayesian belief networks in ecosystem service modelling. *Environmental Modelling & Software*, *46*, 1–11. doi:10.1016/j.envsoft.2013.03.011

Marques, R. L., & Dutra, I. (2008). Redes Bayesianas : o que são , para que servem , algoritmos e exemplos de aplicações. Retrieved from http://pt.scribd.com/doc/3837765/Bayesianas

Mello, M. P., Risso, J., Atzberger, C., Aplin, P., Pebesma, E., Vieira, C. A. O., & Rudorff, B. F. T. (2013). Bayesian Networks for Raster Data (BayNeRD): Plausible Reasoning from Observations. *Remote Sensing*, *5*(11), 5999–6025. doi:10.3390/rs5115999

Mello, M. P., Rudorff, B. F. T., Adami, M., Rizzi, R., Aguiar, D. A., Gusso, A., & Fonseca, L. M. G. (2010). A simplified Bayesian Network to map soybean plantations. In *2010 IEEE International Geoscience and Remote Sensing Symposium (IGARSS 2010)* (pp. 351–354). Honolulu, HI, USA: IEEE. doi:10.1109/IGARSS.2010.5651814

Neapolitan, R. E. (2003). *Learning Bayesian Networks* (p. 674). Pearson Prentice Hall.

Oniśko, A., & Druzdzel, M. J. (2013). Impact of precision of Bayesian network parameters on accuracy of medical diagnostic systems. *Artificial Intelligence in Medicine*, *57*(3), 197–206. doi:10.1016/j.artmed.2013.01.004

Pearl, J., & Russell, S. (2000). Bayesian Networks. *Department of Statistics Papers*. Departament of Statistics, UCLA. Retrieved from https://escholarship.org/uc/item/53n4f34m

Qin, D., Jianwen, M., & Yun, O. Y. (2006). Remote sensing data change detection based on the CI test of Bayesian networks. *Computers & Geosciences*, *32*(2), 195–202. doi:10.1016/j.cageo.2005.06.012

R Core Team. (2014). R: A Language and Environment for Statistical Computing. Vienna, Austria: R Foundation for Statistical Computing. Retrieved from http://www.r-project.org/

Scutari, M. (2009). Learning Bayesian Networks with the bnlearn R Package. *Journal of Statistical Software*, *35*(3), 22. Machine Learning. Retrieved from http://arxiv.org/abs/0908.3817

Suk, H.-I., Sin, B.-K., & Lee, S.-W. (2010). Hand gesture recognition based on dynamic Bayesian network framework. *Pattern Recognition*, *43*(9), 3059–3072. doi:10.1016/j.patcog.2010.03.016

Uusitalo, L. (2007). Advantages and challenges of Bayesian networks in environmental modelling. *Ecological Modelling*, *203*(3-4), 312–318. doi:10.1016/j.ecolmodel.2006.11.033