Bridge Maintenance, Safety, Management, Digitalization and Sustainability –
Jensen, Frangopol & Schmidt (eds)

Fast bridge health monitoring deployment with transferable expertise

R. Sojo-García
Universidad Politécnica de Madrid, Madrid, Spain
Aingura IIoT, Donostia-San Sebastian, Spain

P. Larrañaga & C. Bielza
Universidad Politécnica de Madrid, Madrid, Spain

J. Díaz-Rozo
Aingura IIoT, Donostia-San Sebastian, Spain

ABSTRACT: Machine learning stands out as one of the most widely researched fields in the industry, where it has demonstrated great abilities to model complex system dynamics. This paper takes advantage of one of those models, Bayesian networks (BNs), for analyzing such dynamics in the infrastructure field. Specifically, bridge health monitoring tasks, which are crucial to prevent from dangerous changes within the structure's behavior and ensure the durability of the bridge. However, such modeling requires to capture a substantial amount of data, which can be time-consuming. Considering this challenge, this work aims to accelerate the deployment of these models through related transferable expertise. This involves leveraging data from similar systems to facilitate the learning process of the network. Specifically, two methods, PC-MAX-TL and DBLogLP, are proposed for estimating the structure and parameters, respectively, of the BN under such settings. PC-MAX-TL is an adaptation of the PC-MAX method (a well-known constrained-based structural learning algorithm) tailored to incorporate auxiliary knowledge, while DBLogLP is based on log-linear pooling principles. To demonstrate the reliability of PC-MAX-TL and DBLogLP, an empirical evaluation will be conducted using data from a real scenario: the San Mamés bridge in Bilbao (Spain). This bridge is composed of six different spans that are evaluated independently, and therefore, can be used to address the data scarcity of a certain target span. Then, the results will be compared against the model estimated just with PC-MAX from two perspectives: graphically, looking at the conditional dependencies found in the BNs; and quantitatively, analyzing (1) the network's mean log-likelihood, and (2) the log-likelihood variance. Thus, we demonstrate the proposed methodology's effectiveness, and justify for this bridge that we can accelerate in some days the deployment of the machine learning-based monitoring system.

1 INTRODUCTION

The construction of infrastructures is one of the ancient fields of human development, for which nowadays there are different devices designed to monitor their health evolution along time. Either in industrial environments or for the daily use of people, these systems are thought to withstand extremely variable loads. Therefore, it is important to carefully sensorize them to avoid changes that could affect the endurance of the structure. Since it is impossible to prevent from all kinds of hazards, the usual way of proceeding is to model the behavior of the system and use machine learning techniques to detect deteriorations, thus performing anomaly detection. When these anomalies are spread over time, the infrastructure's model can experience concept drifts, causing it to no longer behave as it used to. One way of modeling that behavior is through BNs, which allow to represent the dependencies of variables

DOI: 10.1201/9781003483755-178

according to the dynamic response of the system and return a measure of how well the model fits the incoming data. However, a correct analysis depends on the collection of sufficient data, something that could require days or even weeks. This problem can be addressed fusing networks of similar characteristics to accelerate the deployment of the model during the earliest stages of data acquisition. Nevertheless, this solution does not consider the particularities of the new data that, indeed, affects the dynamic response. Two methods, PC-MAX-TL and DBLogLP, are proposed for estimating the structure and parameters of the model that represents the incoming data through transferable expertise based on the similarity of other infrastructures with respect to the model of interest. Then, the proposed methodology will be tested with data from a real scenario, the San Mamés bridge in Bilbao (Spain), and evaluated through different analyses against a model obtained with conventional methods.

The paper is organized as follows. Section 2 provides the necessary knowledge about the different types of BNs covered throughout this paper. Section 3 reviews relevant state-of-the-art approaches related to the fusion and construction of networks as well as other transfer learning techniques. Section 4 details the proposed methodology. Section 5 describes the data, and analyzes the results obtained. Finally, Section 6 concludes the paper.

2 THEORETICAL FRAMEWORK

A BN is a machine learning model denoted as $\mathcal{B} = (\mathcal{G}, \theta)$ that comprises a direct acyclic graph (DAG) $\mathcal{G}$ and a set of parameters θ for which $\mathcal{G} = (V, A)$ is defined as a set of nodes V, and a set of arcs $A \subseteq V \times V$ between these nodes. When these arcs are not oriented, they are called edges E. For learning $\mathcal{B}$ we have a dataset $\mathcal{D}$ with n different variables $\mathcal{X} = \{X_1, \ldots, X_n\}$ and N instances $\mathcal{D} = \left\{x^{(1)}, \ldots, x^{(N)}\right\}$. Each arc in A represents a probabilistic dependence between variables, therefore, for each DAG there is a set of conditional dependences and independences embedded in the parameters θ for the conditional probability distribution (CPD) associated to each variable. Then, the joint probability distribution (JPD) of $X_1, \ldots, X_n$ can be factorized using these CPDs. For continuous variables each CPD can be seen as a conditional probability density function (PDF), that can be defined such that:

$$f(X) = \Pi_{i=1}^{n} f(x_i | X_{Pa(i)}) \tag{1}$$

where $\boldsymbol{Pa(i)}$ denotes the parents of X_i in $\mathcal{G}$. We can classify continuous BNs into three main types depending on the assumptions made about the data distribution. These types are parametric, non-parametric and semiparametric BNs.

Parametric BNs assume that the BN can be modeled using a finite number of parameters, where the most popular is the linear Gaussian Bayesian network (LGBN) (Shachter & Kenley, 1989). This type of model assumes that each conditional PDF can be modeled using a Gaussian distribution, for which the mean is a linear combination of some regression coefficients β associated to each node and its parents, and a variance σ^2. On the other hand, non-parametric BNs do not assume any specific distribution for the data. Consequently, there is an unknown, potentially infinite number of parameters in the model. Considering this scenario, Hofmann & Tresp (1996) introduced a non-parametric type based on kernel density estimation (KDE) called KDE BNs. Here, the PDF is estimated taking into account the contribution of each individual instance, becoming a more flexible, but demanding solution, that depends on the training data size and the correct selection of the bandwidth matrix, which acts as a smoothing parameter. Finally, the semiparametric BNs (SPBNs) (Atienza, et al., 2022) encompass both types using Linear Gaussian (LG) CPDs to model parametric relationships, and conditional KDEs (CKDEs) to model the non-parametric ones.

Similarly, there are also different ways of estimating a BN, for which the structure of the data and its corresponding parameters are learnt. There are three main structural learning approaches: constrained-based, which perform conditional independence (CI) tests to assess whether two subsets of variables are conditionally independent given another subset, for instance the PC family methods (PC (Spirtes & Glymour, 1991, PC-Stable (Colombo & Maathuis, 2014) or PC-MAX

(Ramsey, 2016); scored-based, where the best structure is found through an optimization algorithm such as in hill-climbing (HC) (Bouckaert, 1994; and hybrid, where usually the skeleton of the graph is first found through constrained-based methods, and then a scored-based algorithm is used to orientate the edges. Then, depending on whether the CPDs of the nodes are Gausssian or not, the parameters can be obtained through different methods. For this work, we will use the standard maximum likelihood estimate (MLE) or the normal reference rule extended to the multivariate case (Wand, 1992) for bandwidth estimation, respectively.

3 RELATED WORK

Transferring expertise from related problems can be helpful for addressing challenges such as data scarcity or time constraints. This way, we can facilitate the learning process of a new BN taking advantage of the acquired knowledge. However, before exploring the different technologies related to the use of transferable expertise, it is crucial to have some knowledge about the fusion of networks within a specific domain. This process involves the fusion of structures and parameters during the construction of the new network.

Two popular techniques for the fusion of parameters are the *linear pooling* (Stone, 1961) and the *log-linear pooling* (Genest, 1984), for which a weighted arithmetic average and a geometric average are used, respectively, to aggregate the parameters of k models. In terms of structures, the first approach was made in Matzkevich & Abramson (1992) with FUSE_-DAGS. Here, given two DAGs $\mathcal{G}_1$ and $\mathcal{G}_2$, the fusion of structures is performed as the union of their nodes and intersection of their arcs, i.e., $\mathcal{G}_3 = (V_1 \cup V_2, A_1 \cap A_2)$. On the other hand, since the structural fusion usually requires a common topological order among the nodes, two methods for transforming a DAG $\mathcal{G}$ into another given a topological order of nodes were introduced in Matzkevich & Abramson (1993). However, Peña (2011) later critiqued these methods and proposed a set of corrections as well as a complexity analysis. Alternatively, Puerta, et al. (2021) proposed a transformation algorithm and a heuristic search to identify and establish the optimal order, whereas Feng, et al. (2014) introduced a set of desirable properties to any DAG, like avoiding cycles and preserving the conditional independencies. Considering these properties, they proposed a set of addition and deletion rules for the structure learning together with a fuzzy fusion (Liu & Song, 2001) for the parameter aggregation.

Moving to the transferable expertise, we will refer to the model of interest as the target $\mathcal{G}_0$, and the K models for which we have enough data as the auxiliary tasks $\mathcal{G} = \{\mathcal{G}_1, \ldots, \mathcal{G}_K\}$. One of the most important contributions to the state of the art is the work conducted in Luis et al. (2010), where they proposed a modification of the PC algorithm, PC-TL, that linearly combines the information provided by different tasks with the target. This combination is performed considering just the most similar task on each iteration of the learning process. Additionally, they proposed two methodologies for aggregating the parameters, the distance based linear pool (DBLP) and the local linear pool (LoLP). Jia et al. (2010) extended this approach to a soil fertility grading issue by considering, not only the similarity, but also the geographical position of land squares, where it adjusts the weight factors of the linear combination based on the geographical position. Recent contributions such as Hou, et al. (2022) introduce a varying coefficient for parameter learning that adjusts the dependence on the target as the available data increases, whereas Hao et al. (2021) proposed an updating BN model for virtual machines performance estimation, where the structure and parameters are updated according to a re-weighting of the data based on the model performance.

4 METHODOLOGY

The structure learning process of this method is mainly constrained-based although it additionally uses HC to eliminate possible cycles and orient the remaining edges. We refer to it as the PC-MAX-TL algorithm. PC-MAX-TL has been adapted to work without any constraint with both Gaussian and non-Gaussian variables, for which LGBNs and the Fisher's Z test are used in the Gaussian case, and SPBNs and the conditional correlation independence (CCI) test (Ramsey, 2014) in the non-Gaussian. In addition, we present the distance-based log-linear pooling (DBLogLP), which is an aggregation method based on DBLP for the fusion of parameters. This novel aggregation method has the advantage of preserving the conditional independencies and addresses the fusion of PDFs for both Gaussian and non-Gaussian domains. The steps that describe the complete learning process are the following:

1. PC-MAX phase.
   a. Global similarity estimation.
   b. PC-Stable adjacency phase.
   c. Max p-value orientation.
2. HC phase.
   a. Generation of a new fused dataset $\mathcal{D}_f$ for the edge orientation.
   b. Cycle detection. If no cycle is detected use HC to orient the edges, otherwise handle the conflicting arcs and then orient.
   c. Return the best DAG.
3. MLE and DBLogLP for aggregating the target and the auxiliary tasks parameters.

4.1 *PC-MAX-TL*

PC-MAX-TL can be seen as a hybrid learning method with strong constraints into the search space of HC. First, it uses a constrained-based approach that takes advantage of the proposed methodology for PC-TL and one of the algorithms from the PC family, PC-MAX, to analyze the conditional dependencies of the graph. Then, a second phase uses the score and search method HC to address typical problems of this kind of algorithms.

Among the contributions proposed by PC-TL to find a better $\mathcal{G}_0$, there is a metric to measure the similarity S_k between a given auxiliary task and the target. This metric is calculated as the product of a global S_{g_k} and a local S_{l_k} similarities, and used to get the task with the highest value, i.e. the closest task, during the PC adjacency phase. These similarities are defined as:

$$S_{g_k} = dep_k + ind_k, S_{l_k}(X, Y|S) = \begin{cases} 1 \ if \ I_0(X, Y|S) = I_k(X, Y|S) \\ 0.5 \ if \ I_0(X, Y|S) \neq I_k(X, Y|S) \end{cases} \tag{2}$$

$$S_k(X, Y|S) = S_{g_k} \cdot S_{l_k}(X, Y|S) \tag{3}$$

where dep_k and ind_k are respectively, the number of common conditional dependencies and independencies between $\mathcal{G}_k$ and $\mathcal{G}_0$, and $I_0(X, Y|S)$ and $I_k(X, Y|S)$ are the statistics of their CI tests for each pair of X and Y variables conditional on a subset S. Bearing this in mind, $I_F(X, Y|S)$ is defined as the linear combination of the target with its closest auxiliary task:

$$I_F(X, Y|S) = \alpha_0(X, Y|S) \cdot abs(I_0(X, Y|S)) + \alpha_k(X, Y|S) \cdot abs(I_k(X, Y|S)) \tag{4}$$

where *abs(I)* denotes the absolute value of the statistic, and α refers to its confidence factor.

4.1.1 *PC-MAX phase*

The similarity evaluation from PC-TL remains the same for PC-MAX-TL (step 1a). However, to evaluate the edge removals and arc orientations during our PC-Stable adjacency phase (step 1b), it is needed a suitable metric to assess the confidence of each continuous CI test in the combined independence measure. To do so, we kept the reasoning used for the confidence factor α in the original PC-TL, and defined a new confidence factor α_0 such that:

$$\alpha_0 = \frac{N_0}{N_0 + N_k} \tag{5}$$

where N_0 is the size of the target dataset $\mathcal{D}_0$ and N_k the size of the most similar auxiliary dataset $\mathcal{D}_k$. This way, the trust on the target test increases proportionally to N_0, resulting in better independence assessments during the earliest stages of the data acquisition and ensuring future convergences. On the other hand, we reformulated Equation (4) to use the p-value of the statistic during the arc evaluation such that:

$$p_F(X,\ Y|S) = \alpha_0 p_0(X,\ Y|S) + (1 - \alpha_0) p_k(X,\ Y|S) \tag{6}$$

where $p_F(X,\ Y|S)$denotes the combined p-value for each pair of X and Y variables conditional on a subset S. Then, the remaining edges are oriented as possible according to PC-MAX with the combined independence measure (step 1c).

4.1.2 *HC phase*

The result of the PC-MAX phase can be a DAG, although most of the times we end up with a partially directed graph (PDG) that occasionally has cycles. To deal with those situations, we defined the algorithm presented in step 2, where the final orientation is conducted using HC with the k-fold cross-validation log-likelihood score function described in Atienza, et al. (2022). Here, it is important to note that the search space of HC is constrained to the undirected edges that were retained from step 1. Consequently, HC is forced to preserve the arcs of $\mathcal{G}_0$. The cycles are handled by removing one combination of conflicting arcs at a time and adding it into the search space of HC, thereby leaving the decision of either reversing or removing the conflicting arc to the algorithm. However, when $\mathcal{D}_0$ does not contain enough instances the results of the final orientation might be poor due to the lack of information. Therefore, we decided to include a 10% data from the two most similar auxiliary tasks selected throughout the previous phase to address this issue. The result is a new fused dataset $\mathcal{D}_f$ that improves the orientation process leading to better arcs. From this loop, we end with a list of possible DAGs of which the most suitable is selected using the mean of the log-likelihood as a measure of the structure goodness.

4.2 *DBLogLP*

After the structure estimation, the parameters are obtained integrating the knowledge from the target and all the auxiliary tasks. To do so, we obtain a set of initial parameters for each dataset using MLE with the estimated DAG from PC-MAX-TL. This way, it is ensured that the parameters of both target and auxiliary tasks come from the same set of parents, otherwise the aggregation with DBLogLP (step 3) would not be possible. The procedure of DBLogLP first calculates a geometric average of auxiliary parameters that are later combined with those of the target, thereby preserving the conditional independencies. The mean PDF for node i is defined as:

$$\bar{f}(X_i) = \prod_{k=1}^{K} f_k(X_i)^{w_{ki}} \tag{7}$$

where w_{ki} is the weight associated to the auxiliary task k for the PDF of node i. This weight is calculated as the normalized inverse of the maximum Kullback-Leibler divergence (KL) between the auxiliary conditional PDF $f_k(X_i)$ and the target conditional PDF $f_0(X_i)$ (Koliander et al., 2022). Then, the resulting parameters from Equation (7) are integrated with those of the target in the following way:

$$f_0'(X_i) = f_0(X_i)^{c_N} \bar{f}(X_i)^{1-c_N} \tag{8}$$

where c_N is the confidence coefficient that indicates how much to trust the estimations of the target. Similarly to α_0 in Equation (5), this coefficient depends on the number of instances from the target. Therefore, it is obtained as:

$$c_N = \frac{N_0 - N_{min}}{N_{max} - N_{min}},\ 0.25 \le c_N \le 0.75 \tag{9}$$

where N_{min} and N_{max} are the normalization factors that must be defined at the beginning of the algorithm. Here, we decided to limit the values of the confidence coefficient c_N to keep the trust in the target parameters between the 25% and 75%. Thus, neither the target is completely ignored, nor the auxiliary tasks. It must be noted that depending on the type of BN, the parameters to be combined are different. For Gaussian nodes, we will have to aggregate LG

CPDs whose parameters are the variance σ^2 and the regression coefficients β, while for non-Gaussian nodes we have CKDEs and the bandwidth matrix.

5 EXPERIMENTS

We will use data from our Aingura Insights Edge device deployed at the San Mamés bridge in Bilbao, Spain. The bridge incorporates a mixed metallic structure with V-shaped supports and consists of six different spans whose characteristics are slightly different, allowing us to categorize them as different structures within a common system (the bridge). The total length of the bridge encompasses 323 m, but each span has a specific length. These lengths are 53 m for spans 1 and 5, 60 m for spans 2, 3 and 4, and 37 m for span 6. Each span has its corresponding dataset, which is composed by different pairs of natural frequencies and damping factors estimated from accelerometers readings by proprietary algorithms deployed at the edge device. These variables capture the dynamic response of the bridge, expressed through natural frequencies characterized by mixtures of Gaussians and damping factors following Gamma distributions. Anyhow, we will evaluate the methodology under both Gaussian and non-Gaussian circumstances. The data used for this purpose was collected between the 23rd of March and the 3rd of May 2023, which represents approximately 1000 vehicles per day. We run one experiment per target span, while the five remaining spans are use as auxiliary data. With respect to the configuration of the experiments, we have selected different significance level for each CI test associated to each BN. In the case of LGBNs and the Fisher's Z test, we use a 0.05 significance level, however, experimentally we found that this level was too restrictive for the CCI test. Hence, in that case, we incremented the value from 0.05 to 0.08. Additionally, N_{min} and N_{max} are set to 20 and 1500 in both LGBNs and SPBNs. Finally, we have selected the first 3000 instances for the training sets of the auxiliary data, whereas the target sample is gradually incremented in batches of 40, starting in 20 until 2000. The remaining data is saved for testing.

That said, the first thing we can notice when we draw the DAG of the standard PC-MAX (Figure 1), is the inability of the algorithm to find relationships between the nodes at the earliest stages of the data acquisition. Besides, the SPBN is unable to correctly assess the type of the nodes in the DAG, where the white nodes are LG CPDs and the grey nodes CKDEs. Here, the natural frequencies 3 and 5, and the damping factors 2 and 4, were set as Gaussian nodes, when as we mentioned earlier they look like mixtures of Gaussians and Gamma distributions, respectively. On the contrary, it can be seen from Figure 2 that PC-MAX-TL is able to address this issue quite effectively. In fact, it correctly captures the dependencies of the natural frequencies establishing arcs from the first two to the third, fourth and fifth, while each of them is paired with its damping factor both in LGBNs and SPBNs (NAT_FREQ and DAMP in the graph).

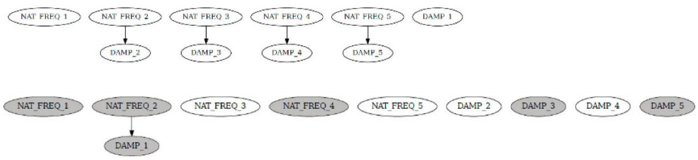

Figure 1. PC-MAX DAG from span 1 with 20 instances. Top: LGBN. Bottom: SPBN.

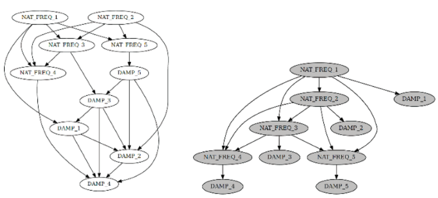

Figure 2. PC-MAX-TL DAG from span 1 with 20 instances. Left: LGBN. Right: SPBN.

As regards the quality assessment of the network in terms of mean log-likelihood (Figure 3), the PC-MAX-TL models (blue lines) exhibit better starting points in general. Then, they maintain the score within a small range of variation, whereas the network alone (PC-MAX) needs some time to reach such level. The convergence of both systems depends on the model, for instance, in span 1 it happens around 1000 and in span 6 it is almost immediately, but around 500 instances most of them have converged or reduced significantly the difference. However, we also see some drops in the means of spans 1 and 5 within the SPBN, as well as span 6 in both LGBN and SPBN. In this case, the issue may lie within the training set. The inclusion of certain portions of noisy data during experiments could be contributing to an increase in the system's variability, consequently resulting in a drop in the score. On the other hand, the SPBNs seem to be more sensitive to this noise than the LGBNs due to their parameter aggregation methodology, which requires the fusion of bandwidth matrices, thereby affecting to the smoothing of PDFs in the CKDE nodes.

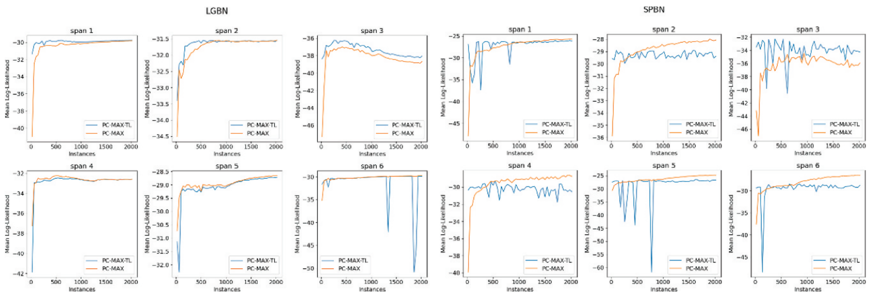

Figure 3. Networks mean log-likelihood. Left: LGBN. Right: SPBN.

Finally, the stability analysis (Figure 4) shows that the deviations of the PC-MAX-TL models (blue lines) exhibit very few variations from the very beginning, returning more stable measures with lower deviations in practically all cases from both LGBNs and SPBNs. In fact, even in the scenarios where the deviation of PC-MAX is lower, the stability tends to converge to the level of PC-MAX-TL, that is the case of spans 2 and 4. Additionally, it can be seen that the drops in the means detailed in the previous paragraph match the peaks in the deviations, which as mentioned, is caused by an increase in the variability of the data. However, even under the presence of noise their stability levels are higher than in PC-MAX. For instance, the deviations of spans 1 and 3 in the SPBN are 10 to 20 units lower.

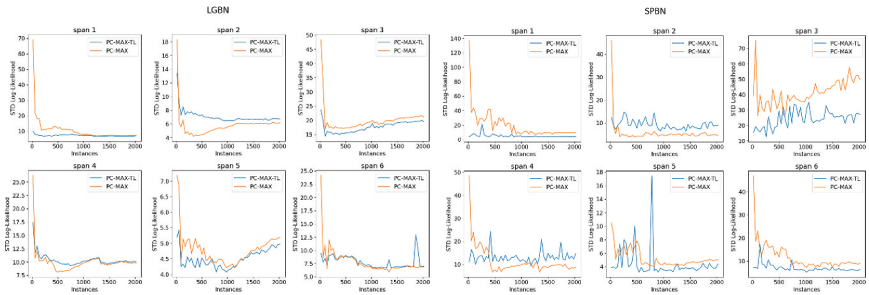

Figure 4. Networks stability. Left: LGBN. Right: SPBN.

6 CONCLUSION

The use of transferable expertise has been the basis of our methodology composed by two novel procedures, PC-MAX-TL and DBLogLP, for which we have reviewed all the aspects involved in the network construction and justify their reliability for building and aggregating models in presence of data scarcity. PC-MAX-TL returns DAGs with much better conditional dependencies compared to PC-MAX at the earliest stages of data acquisition, and its combination with DBLogLP ensures a much more stable model despite the noise, which is crucial for its application in the industry. The convergences achieved in terms of mean log-likelihood are around 500 instances, while the stability returns better scores throughout the whole experiment. Therefore, considering that on average the San Mamés bridge gets 1000 impulses per day, the most important impact is that the required time for the bridge health monitoring systems could be reduced by one or two days.

ACKNOWLEDGEMENTS

This work has been partially supported by the Spanish Ministry of Science and Innovation through the RTC2019-006871-7 DSTREAMS project, the PID2022-139977NB-I00 and TED2021-131310B-I00 projects, and by the grant to the ELLIS Unit Madrid by the Autonomous Region of Madrid.

REFERENCES

Atienza, D., Bielza, C., & Larrañaga, P. (2022). Semiparametric Bayesian networks. *Information Sciences*, 564–582.

Bouckaert, R. R. (1994). Properties of Bayesian belief network learning algorithms. *Proceedings of the Tenth Conference on Uncertainty in Artificial Intelligence*, 102–109.

Colombo, D., & Maathuis, M. H. (2014). Order-independent constraint-based causal structure learning. *Journal of Machine Learning Research*, 3921–3962.

Feng, G., Zhang, J.-D., & Liao, S. S. (2014). A novel method for combining Bayesian networks, theoretical analysis, and its applications. *Pattern Recognition*, 2057–2069.

Genest, C. (1984). A characterization theorem for externally Bayesian groups. *The Annals of Statistics*, 1100–1105.

Hao, J., Gan, J., Zhou, J., & Wang, J. (2021). Performance prediction of virtual machines via transfer learning of Bayesian network. *2021 IEEE 6th International Conference on Computer and Communication Systems*, 217–222.

Hofmann, R., & Tresp, V. (1996). Discovering structure in continuous variables using Bayesian networks. *Advances in Neural Information Processing Systems 8*.

Hou, Y., Yang, A., Guo, W., Zheng, E., Xiao, Q., Guo, Z., & Huang, Z. (2022). Bearing fault diagnosis under small data set condition: A Bayesian network method with transfer learning for parameter estimation. *IEEE Access*, 35768–35783.

Jia, H.-Y., Chen, J., Yu, H.-L., & Liu, D.-Y. (2010). Soil fertility grading with Bayesian network transfer learning. *2010 International Conference on Machine Learning and Cybernetics*, 1159–1163.

Koliander, G., El-Laham, Y., Djurić, P. M., & Hlawatsch, F. (2022). Fusion of probability density functions. *Proceedings of the IEEE*, 404–453.

Liu, W.-Y., & Song, N. (2001). The fuzzy association degree in semantic data models. *Fuzzy Sets and Systems*, 203–208.

Luis, R., Sucar, L., & Morales, E. (2010). Inductive transfer for learning Bayesian networks. *Machine Learning*, 227–255.

Matzkevich, I., & Abramson, B. (1992). The topological fusion of Bayes nets. *Uncertainty in Artificial Intelligence*, 191–198.

Matzkevich, I., & Abramson, B. (1993). Deriving a minimal I-map of a belief network. *Uncertainty in Artificial Intelligence*, 159–165.

Peña, J. M. (2011). Finding consensus Bayesian network structures. *Journal of Artificial Intelligence Research*, 661–687.

Puerta, J. M., Aledo, J. A., Gámez, J. A., & Laborda, J. D. (2021). Efficient and accurate structural fusion of Bayesian networks. *Information Fusion*, 155–169.

Ramsey, J. D. (2014). A scalable conditional independence test for nonlinear, non-Gaussian data.

Ramsey, J. D. (2016). Improving accuracy and scalability of the PC algorithm by maximizing p-value.

Shachter, R. D., & Kenley, C. R. (1989). Gaussian influence diagrams. *Management Science*, 527–550.

Spirtes, P., & Glymour, C. (1991). An algorithm for fast recovery of sparse causal graphs. *Social Science Computer Review*, 62–72.

Stone, M. (1961). The opinion pool. *The Annals of Mathematical Statistics*, 32(4) 1339–1342.

Wand, M. P. (1992). Error analysis for general multivariate kernel estimators. *Journal of Nonparametric Statistics*, 1–15.