Bayesian Gaussian networks for multidimensional classification of morphologically characterized neurons in the NeuroMorpho repository

P. Fernández-González, P. Larrañaga, C. Bielza

Department of Artificial Intelligence
Technical University of Madrid
{pablo.fernandezgonz;pedro.larranaga;mcbielza}@fi.upm.es

Abstract. A class-bridge decomposable multidimensional Gaussian network is presented as an interpretable and powerful model, to account for the morphological differences that exist between different neurons when varying the species, gender, brain region, cell types and developmental stage of the animal of origin. Also this work includes a learning algorithm that makes use of the CB-decomposablility property to alleviate the inference complexity and use it to learn complex network structures that take into account relationships between classes. The model is trained with data from NeuroMorpho (v5.7) and the final model is used to test the predictive power of the learning algorithm for Bayesian networks and, given its interpretability, to extract knowledge at a neuroscience level.

1 Introduction

Neurons's morphology differences have been observed between different animals, but also within the same species. The developmental stage and the location in the brain can also show morphological variations between cells [5]. In order to statistically analyze these differences, a multidimensional classifier using an interpretable statistical model is one of the most appealing approaches. To build a model that can effectively predict class labels such as in which specie, gender and developmental stage an animal is and to which cell types the sample neuron belongs, given a set of morphological descriptors of the neuron, could be considered a big step towards neuron morphology understanding. Specially if the selected model has the property of interpretability, allowing us to extract knowledge directly from it.

For this work, a class-bridge decomposable multidimensional Gaussian classifier (CB-MGC) is proposed and trained with the neurons dataset [1]. This classifier is not bounded to a prefixed structure (naive Bayes, tree-like structures in the class variables, etc.) and also handles variables of continuous (Gaussian) and discrete nature. Its influenced by the works of [9] and [3]. The classifier's strengths are its interpretability, the capability to capture dependencies between the class variables, the exploitation of the class-bridge decomposability property and its ability to handle feature variables of continuous nature straightforwardly, without the need to discretize the data. Its weakness may be the assumption of

Actas de la XVII Conferencia de la Asociación Española para la Inteligencia Artificial, pp. 39-48

Gaussianity in the continuous nodes, where features whose distribution strongly deviates from the Gaussian distribution could hinder the model's performance. However, this is acceptable in this case as the data features tend to distribute according to Gaussian distributions. The definition and properties of this model will be detailed in Section 2.

Section 3 presents a structural learning algorithm that uses the class-brigde decomposability to incrementally build a complex network structure while saving computational costs in the process.

Section 4 shows the results and final network with a focus on the implications of the obtained relationships in the final model. Finally, Section 5 summarizes the main findings and discusses the conclusions and implications of this work.

2 Multidimensional Gaussian network classifiers

A Bayesian network (BN) is a pair $\mathcal{B} = (\mathcal{G}, \Theta)$ over a set of random variables $\chi = \{X_1, X_2, \ldots, X_n\}$ where $\mathcal{G} = (V, A)$ is a Directed Acyclic Graph (DAG) with a set of vertices V and a set of arcs A and Θ is a set of probability distributions associated with the random variables in χ. Vertices represent the variables in χ and the directed arcs represent probabilistic dependence relationships between the variables. Probability distributions in Θ satisfy $\theta_{x_i|\mathbf{pa}(x_i)} = p(x_i|\mathbf{pa}(x_i))$, that is, conditional probability distributions of variable X_i given a value of the set of variables $\mathbf{Pa}(X_i) \in \chi$. In here $\mathbf{Pa}(X_i)$ stands for the set of parent variables of X_i in $\mathcal{G}$. Bayesian networks factorize a joint probability distribution as follows:

$$p(X_1, \ldots, X_n) = \prod_{i=1}^{n} p(X_i|\mathbf{Pa}_{(X_i)}) \tag{1}$$

A multidimensional Gaussian network classifier (MGNC) is a Bayesian network over a set $\chi_c = \{X_1, \ldots, X_m\}$ of continuous random variables and a set $\chi_d = \{C_1, \ldots, C_{n-m}\}$ of discrete class random variables where χ_c is assumed to be jointly distributed as a multidimensional Gaussian distribution $\mathcal{N}(\mu, \sum)$, where μ is a vector of means and $\sum$ is the covariance matrix of the variables in χ_c. χ_c and χ_d are referred to as the set of feature variables and the set of class variables, respectively. MGNCs are additionally constrained to satisfy $\mathbf{Pa}(C_i) \cap \chi_c = \emptyset$, that is, no arcs from feature variable to class variables are permitted. Multidimensional classifiers have been studied initially in [11], and extended in [2] and [3].

In concordance with the literature, MGNC can be additioanlly described by considering three different subgraphs in its structure:

- $A_C \subseteq V_C \times V_C$ is the set of arcs connecting solely the class variables. The associated subgraph, that contains as nodes all the class variables and is induced by V_C, is denoted as $\mathcal{G}_C = (V_C, A_C)$
- $A_X \subseteq V_X \times V_X$ is the set of arcs connecting solely the feature variables. The associated subgraph, that contain as nodes all the feature variables and is induced by V_C, is denoted as $\mathcal{G}_X = (V_X, A_X)$
- $A_{CX} \subseteq V_C \times V_X$ is the set of arcs that go from the class variables to the features variables. The associated subgraph comprehends all nodes of the network as is denoted as $\mathcal{G}_{CX} = (V, A_{CX})$

For this type of models, classification using a 0-1 loss function amounts to solving the most probable explanation (MPE) problem. That is, for an instance of the feature variables $\mathbf{x} = (x_1, \ldots, x_m)$ the problem can be written as:

$$\mathbf{c}^* = (c_1^*, \ldots, c_n^*) = arg \max_{c_1,\ldots,c_n} p(C_1 = c_1, \ldots, C_n = c_n|\mathbf{x}) \tag{2}$$

That is, the search of the class labels that maximizes the probability of the class variables given the evidence of the feature variables. When calculating the MPE in a MGNC, its possible to use Equation (1) to compute it by considering $p(c|\mathbf{x}) \propto p(c, \mathbf{x})$ where $p(c_i|\mathbf{pa}(c_i))$ is computed as a classical discrete probability in a BN and for the feature nodes, $f(x_i|\mathbf{pa}(x_i)$ follows a Gaussian distribution $\mathcal{N}(\mu_i, v_i)$ where

$$\mu_i = \mu_{i|pc_i} + \sum_{j=1}^{n_i} \beta_{ij|pc_i}(x_j - \mu_{j|pc_i})$$

$$v_i = \frac{\left|\sum_{X_i,PX_{i|pc_i}}\right|}{\left|\sum_{PX_{i|pc_i}}\right|}$$

where $pc_i = \mathbf{pa}_{V_{C_i}}(x_i)$ is the set of class parents of X_i, n_i is the number of feature parents of X_i, $\beta_{ij|pc_i}$ is a regression coefficient defined as:

$$\beta_{ij|pc_i} = \frac{\sigma_{ij|pc_i}}{\sigma^2_{j|pc_i}} \tag{3}$$

and $\sum_{L|pc_i}$ is the covariance matrix of the set of variables L conditioned to the class parents of X_i.

A Gaussian network possesses several desired properties such as the less demanding number of parameters to model a continuous distribution ($\mathcal{O}\left(\prod_{i=1}^{m} r_i\right)$ where $r_1, \ldots, r_m$ are the cardinalities of the variables $X_1, \ldots, X_m$ in the discrete case vs. $\mathcal{O}(n^2 r)$ in the continuous one) and the possibility to compute them independently from the structure of the GN [4] ($\mathcal{O}(.)$ is used as an indicator of complexity).

The computation of the MPE, however, concerns only the class variables, that is the discrete part of the network, and therefore no complexity alleviation was found for inference by assuming gaussianity in the feature nodes. This is a well-known problem as when learning an unrestricted class structure the MPE problem is exponential in the number of variables. Which renders the inference intractable for a relatively small set of class variables.

2.1 Class-bridge decomposability property

In order to tackle the inference problem, CB-decomposable MGNCs are considered, extending previous works [2] and [3] for discrete feature variables. A MGNC is a CB-decomposable MGNC if it satisfies the following two properties:

– $\mathcal{G}_C \cup \mathcal{G}_{CX}$ can be partitioned as $\mathcal{G}_C \cup \mathcal{G}_C X = \bigcup_{i=1}^{r}(\mathcal{G}_{C_i} \cup \mathcal{G}_{CX_i})$, where $\mathcal{G}_{C_i} \cup \mathcal{G}_{CX_i}$, for $i = 1, \ldots, r$ are subsets of the original graph denoted as r maximal connected components.
– $Ch(V_{Ci}) \cap Ch(V_{C_i}) = \emptyset$ with $i, j = 1, \ldots, r$ and $i \neq j$, where $Ch(V_{C_i})$ stands for the set of children variables of V_{C_i}. The subset of class variables in $\mathcal{G}_{C_i}$ (i.e non-shared children property).

Then the MPE problem for a CB-decomposable MGNC is transformed into

$$arg \max_{c_1,\ldots,c_n} p(C_1 = c_1, \ldots, C_n = c_n|x) \tag{4}$$
$$\propto \prod_{i=1}^{r} \max_{\mathbf{c}^{\downarrow V_{C_i \in I_i}}} \left(\prod_{C \in CV_{C_i}} p(c|\mathbf{pa}(c)) \prod_{X \in Ch(V_{C_i})} p(x|\mathbf{pa}_{V_C}(x), \mathbf{pa}_{V_X}(x))\right)$$

where $\mathbf{c}^{\downarrow V_{C_i \in I_i}}$ is the projection of the vector $\mathbf{c}$ to the coordinates in V_{C_i} and I_i stands for the sample space associated with V_{C_i}. Intuitively, this breaks the MPE problem into r smaller MPE problems. Given the exponential nature of the total of possible label combinations w.r.t. the number of class variables, this effectively alleviates the computational burden as well as the sample size needed for the classification problem. Its also possible to see this property in the factorization of the network, as each component is identified as a subset of the network factors whose class variables form a closed group (that is, no other reference is found to them in the rest of the factors of the network).

3 Structural learning algorithm

The proposed learning algorithm can be characterized as a 3-step learning algorithm with a greedy forward search approach. That is, arcs are initialized to the empty set for the three different subgraphs $A_C = \emptyset, A_X = \emptyset$ and $A_{XC} = \emptyset$, obtaining an initial network with no arcs and all nodes present. Then, it follows with the addition of arcs to the different parts of the network judging their contributions using the global accuracy criteria and exploiting the CB-decomposable property to escalate to complex network structures without too much computational burden, aiming to obtain a sufficiently good local optimal structure. Also, this algorithm explores the efficency of introducing feature subset selection strategies when learning the bridgde subgraph, both in a filter and wrapped fashion, to ensure the quality of the first step resulting structure.

3.1 Learning the bridge subgraph

The algorithm first focus on building a naive Bayes subgraph $NB_i(C_i, \chi_e)$, with $\chi_e \subset \chi_c$ for each class variable $C_i, i = 1, \ldots, n$ of the network, over which a sequential feature subset selection process is carried out. First, the features are grouped according to their separation power by means of a Kruskal-Wallis test [6]. Each feature data is partitioned into subgroups according to the class label. Since features with lower $p-$values are considered to be more relevant for classification, they are sorted in ascending value. Then, the sequential feature subset selection

technique is applied, which adds arcs from the C_i variables to X_j variables if an accuracy improvement is detected.

Finally, it eliminates shared children in order to obtain an initial CB-decomposable MGNC structure with the maximum number of r maximal connected components, where $r = n$ since each naive bayes graph is a maximal connected component. In order to do this, it compares the $p-$values obtained in Kruskal-Wallis test for classes C_i and C_l and variable X_j and removes the arc that had a higher associated $p-$value. If equal, arc removal is chosen randomly.

The algorithm is depicted as follows:

1. **for** i $= 1$ **to** $n - m$ **do**
   (a) Select class variable C_i
   (b) Initialize the set of features as $\chi_i = \emptyset$
   (c) **for** j $= 1$ **to** m **do**
      i. Separate feature X_j according to the values of C_i
      ii. Obtain $p-$value from Kruskal-Wallis test
   (d) Sort features according to ascending $p-$values
   (e) **for** j $= 1$ **to** m **do**
      i. **if** $Acc(NB_i(C_i, \chi_i)) < Acc(NB_i(C_i, \chi_i \cup X_j))$ **do**
         A. $NB_i(C_i, \chi_e) := NB_i(C_i, \chi_e \cup X_j)$
2. Compare all the children of all NB_i and for each pair $NB_a(C_a, \chi_a), NB_b(C_b, \chi_b)$ such that $\chi_a \cap \chi_b \neq \emptyset$ do
   (a) Compare $p-$values in all $X_p \in \chi_a \cap \chi_b$ **If**. $kwpval(X_p, C_a) > kwpval(X_p, C_b)$ **do**
      i. Remove arc from C_a to X_p in NB_a
   (b) **else if** $kwpval(X_p, C_a) < kwpval(X_p, C_b)$
      i. Remove arc from C_b to X_p in NB_b
   (c) **else**
      i. Arc removal chosen randomly between 2.1.i and 2.2.i
3. Output $\mathcal{G}_{CX} = \bigcup_{i=1}^{n-m} NB_i$

3.2 Learning the feature subgraph

The second step is to obtain the feature subgraph, for which a maximum number of iterations parameter t, of arc insertions attempts, is defined. This decision was adopted to avoid the computational burden of examining all possible arc insertions. First, the algorithm calculates the global accuracy that corresponds to the concatenation of the individual class predictions of all existing maximal connected components.

Arc insertions may occur from unselected features in the previous process, but only to the features that are part of the components at the moment of the arc insertion. When an arc insertion occurs, the parent feature is added to the component. For each arc insertion between a pair of nodes $X_i \rightarrow X_j$ the accuracy is recalculated. It is important to note that because of the CB-decomposability property, at this step only the MPE values for the class of the component containing the children node need to be recalculated. If there is a global accuracy improvement, the arc insertion is kept, otherwise is discarded. Because accuracy is used as the metric for the arc insertions, this is a wrapper structural learning step.

3.3 Learning the class subgraph

For the final graph, the algorithm tries to identify the existing dependencies between class variables and attempt to merge the r maximal connected components. It does this, like in the previous step, in a wrapper fashion. The algorithm starts by considering all possible pairwise components mergings. For each component, all single arc insertions between classes that belong to different components are evaluated, in both directions. If an improvement in accuracy exists, the arc insertion process continues updating the merged component class subgraph by further arc insertions, this process finishes when no improvement in accuracy is observed. Similarly, the merging components process finishes when no component merging improves accuracy or when the number of components has been reduced to one. It is important to notice that when two components are merging, the MPE values only need to be reevaluated for those two components, leaving the remaining nodes outside. This process of local computations guarantees that the computational burden of the MPE increases exponentially only when an arc insertion produces a network topology that cannot be separated in smaller maximal connected components, and involves a higher number of class variables. If there is only two components and are merging, the MPE is computed similarly to a classic exact inference approach involving all class variables. The algorithm is depicted as follows:

1. Initialize AccImprovement = true, ComponentAccImprovement = true, $\mathcal{R}_c = \{R_1, \ldots, R_n\}$ where each $R_i \in \mathcal{R}_c$ is a GN (initially is the list of components obtained in step 2)
2. **while** AccImprovement and $|\mathcal{R}_c| > 1$ **do**
   (a) $lR := \emptyset$
   (b) **For each** possible R_i, R_j component merging where $i, j = 1, \ldots, n$ and $i \neq j$ **do**
      i. $R_{ij} := R_i \cup R_j$
      ii. $aR_{ij} = R_{ij}$
      iii. **while** ComponentAccImprovement **do**
         A. Evaluate all possible single arc insertions $C_{R_ik} \rightarrow C_{R_jh}$, $C_{R_ik} \leftarrow C_{R_jh}$ from class nodes of different components in R_{ij}
         B. **if** exists arc insertions that improves component accuracy **do** select best arc and update R_{ij}
         C. **else** ComponentAccImprovement = false
      iv. **if** $aR_{ij} \neq R_{ij}$ **do** $lR := lR \cup R_{ij}$
   (c) **if** $lR \neq \emptyset$ **do** select the best merging of components, R_{ab}, contained in lR, $\mathcal{R}_c := \mathcal{R}_c - \{R_a, R_b\}$ and $\mathcal{R}_c := \mathcal{R}_c \cup R_{ab}$
   (d) **else** Accimprovement = false
3. Return the obtained CB-MGC = $\bigcup_{i=1}^{|\mathcal{R}_c|} r_i \in \mathcal{R}_c$

It should be noted that the class sub-graph is not bounded to any network topology or any subset of all possible networks, which itself offers a great appeal with respect to restricted methods. This learning algorithm operates by escalating the complexity of the network topology through a path that minimizes the computational burden of calculating the MPE at each step, by exploiting the CB-decomposability property.

4 Classification of neuron's morphologycall features

The data was obtained from NeuroMorpho v5.7.org, more specifically, the available data from [1]. In its raw form, the dataset contained information about 10880 3D reconstructed neurons, that were later processed with the L-measure tool [10] to extract a total of 215 features describing the neurons morphology. Initially, the dataset was composed of seven class labels (specie, gender, brain region, cell type level 1, cell type level 2, development and neocortex) with missing data, which shows that the initial problem is a multidimensional semisupervised classification problem. Another difficulty was that some class labels were heavily imbalanced, with the most extreme case represented by a Rabbit's neuron, with only one instance for class variable specie. Hence, a preprocessing step was conducted combining data imputation (using a 1-NN nearest neighbors algorithm) with the elimination of class values that did not reach a critical l number of instances (l can be regarded as a parameter to the final model that shapes the data that the learning algorithm receives). This number was set to be $l = 200$. Preprocessing further continued as for the classifier optimize its performance, features must not significantly deviate from gaussianity and data fitting to a Gaussian distribution should be possible under all data subsets originated from conditioning the feature to the class labels. With this, dataset pruning further continued to reach a final count of 5136 instances, 6 classes (the neocortex class variable was left out as most its values were missing) and a total of 158 features (57 were either too different from Gaussian distributions or had subpopulations with zero variance).

A more detailed description of the class labels can be gound in Table 1. They conform a class cardinality space of 1800 possible label combinations.

Table 1: Class labels in the final dataset

Specie	Gender	Brain region	Cell type level 1	Cell type level 2	Develop.
drosophila	female	anterior olfactory nucleus	axonal terminal	ganglion cell	adult
human	male	basal forebrain	interneuron	granulle cell	young
monkey		hippocampus	principal cell	medium spiny cell	
ray		neocortex		pyramidal cell	
		optic lobe		tangential cell	
		retina			

The algorithm is now applied, trainning a CB-decomposable multidimensional classifier with the goal of finding relationships in the data that can help us understand and predict how neuron morphology changes across the different class labels. This algorithm was programmed using Matlab (version R2015a) and the Bayes net toolBox [8] package together with the Structural learning package [7].

As seen in Figure 1, 6 components have been obtained that noticeably differ from each other after the first two steps. The parameter t for arc insertion attempts was fixed at $t = 250$ although it can be observed that most of the arc insertions did not improve the final accuracy of the model and hence only a small subset produced definite arc inclusions. The software L-measure generally reports

Fig. 1: Components after second phase of computation. Learned Feature subgraph.

the minimum, maximum average and standard deviation values as descriptive features of some measured aspect of the neuron. It can be consistently observed thought the components how these values tend to appear together in the components (for example the parent_daughter_ratio that measures the ratio between the diameter of a dendrite or axonal segment and its segment prologations after a bifurcation has taken place) which strengthens understanding of a statistical dependency existing between that measured aspect and the class variable connected to the features that describe it. Its also worth noticing that after computing step 2, the same node can appear in two different components, but as child of the feature variables. When components are merged in phase 3, intersecting features are merged together.

In Figure 2 the found dependencies between classes in the final network, after computing step 3, are visualized. As it seems, the variable for specie represents the major discriminant between the morphological features of two neurons, as it conditions all but the development variable. This supports the common intuition that two animals from different species differ more in their morphology than, for example, two animals of the same specie but of different genders. Along with intuition also seems to be the dependency between brain_region and cell_type_level 2 as different areas of the brain tend to have different neuron subpopulations. The gender dependency by development suggests that morphological differences between individuals of different genders vary with time (intuitively maybe this corresponds to the stages of sexual differentiation in the transition from young to adult that some species experience, or a sexual homogenization passing from

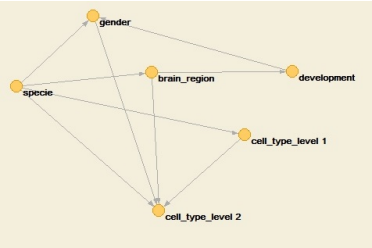

Fig. 2: Final class subgraph depicting the dependencies found in the data

adult to old). Moreover, cell_type_level 2 seems to be the most dependent of all classes, which also seems intuitive as its measured at the smallest granularity, that is "is the closest to an individual neuron" or the one that has potentially less variability. These findings significantly improve the confidence on those previously hypothesized relationships between these classes.

The final model performance as a multidimensional classifier was measured by the Hamming score and global acuracy metrics, and found values 0.7666 and 0.2288, respectively. These metrics are defined, respectively, as follows:

$$H_s = \frac{1}{N}\sum_{i=1}^{N}\frac{|T_i \cap P_i|}{|T_i \cup P_i|} \tag{5}$$

where N is the total number of instances in the data, T_i is the set of true labels for the $i-$th instance and P_i is the set of predicted labels by the classifier for the $i-$th instance.

$$G_a = \frac{1}{N}\sum_{i=1}^{N}\delta_{T_i}^{P_i} \tag{6}$$

where $\delta_{T_i}^{P_i}$ is a function that outputs 1 if $T_i = P_i$ and 0 otherwise. The algorithm was tested using a train/split fashion where $\frac{1}{3}$ of the total dataset instances were used and randomly chosen for testing.

5 Conclusions and future lines of research

This new learning algorithm for a multidimensional classifier effectively models and predicts multiple classes provided a set of features. Also, it cab ne effectively used to build a model that predicts multiple classes of a neuron given a set of morphological descriptors. It is worth noticing that the obtained class subgraph could not have been obtained under common restrictions for multidimensional classifiers, such as independent classes, sequential(chain) dependencies or tree

structures. Therefore, this model offers a superior performance in terms on interpretability. This was achieved by the continuous usage of the CB-decomposability property through the learning process, allowing it to escalate from simple to complex network topology without computing the MPE problem with more variables than necessary. It also succeeds in the objective of extracting useful knowledge out of the data in the field of neuroscience, which we believe validates the application of our model to real life problems and our choice of this model for this problem.

5.1 Improvements and future lines of research

The 1-NN data imputation method will be substituted by a method based on the structural learning process. As in its current form, the structural learning algorithm does not explore the addition of arcs between class variables and feature variables that belonged to different components when merging, an investment in computational power that could lead to significant improvements in the classifiers accuracy. Also, the addition of arc removal operations can be considered.

References

1. Ascoli, G.A., E., D.D., Halavi, M.: NeuroMorpho.org: a central resource for neuronal morphologies. The Journal of Neuroscience 27(35), 9247-9251 (2007)
2. Bielza, C., Li, G., Larrañaga, P.: Multi-dimensional classification with Bayesian networks. International Journal of Approximate Reasoning 52(6), 705–727 (2011)
3. Borchani, H., Bielza, C., Larrañaga, P.: Learning CB-decomposable multi-dimensional Bayesian network classifiers. In: Proceedings of the 5th European Workshop on Probabilistic Graphical Models (PGM'10), pp. 25–32. Helsinki (2010)
4. Geiger, D., Heckerman, D.: Learning Gaussian networks. In: Proceedings of the Tenth international conference on Uncertainty in Artificial Intelligence. pp. 235–243. Morgan Kaufmann Publishers Inc. (1994)
5. Jacobs, B., Johnson, N., Wahl, D., Schall, M., Maseko, B.C., Lewandowski, A., Raghanti, M.A., Wicinski, B., Butti, C., Hopkins, W., Bertelsen, M.F., Walsh, T., Roberts, J.R., Reep, R., Hof, P.R., Sherwood, C.C., Manger, P.: Comparative neuronal morphology of the cerebellar cortex in afrotherians, carnivores, cetartiodactyls, and primates. Frontiers in Neuroanatomy 8 (2014) article 24
6. Kruskal, W.H., Wallis, W.A.: Use of ranks in one-criterion variance analysis. Journal of the American statistical Association 47(260), 583–621 (1952)
7. Leray, P., Francois, O.: BNT structure learning package: Documentation and experiments. Laboratoire PSI, Universitè et INSA de Rouen, Tech. Rep (2004)
8. Murphy, K., et al.: The Bayes net toolbox for Matlab. Computing Science and Statistics 33(2), 1024–1034 (2001)
9. Pérez, A., Larrañaga, P., Leguey, I., Inza, I.: Supervised classification with conditional Gaussian networks: Increasing the structure complexity from naive Bayes. International Journal of Approximate Reasoning 43(1), 1–25 (2006)
10. Scorcioni, R., Polavaram, S., Ascoli, G.A.: L-measure: a web-accessible tool for the analysis, comparison and search of digital reconstructions of neuronal morphologies. Nature Protocols 3(5), 866–876 (2008)
11. Van Der Gaag, L.C., De Waal, P.R.: Multi-dimensional Bayesian network classifiers. In: Probabilistic Graphical Models. pp. 107–114. Prague (2006)