Multidimensional classifiers for neuroanatomical data

Pablo Fernandez-Gonzalez, Concha Bielza, Pedro Larranaga

Multidimensional classifiers for neuroanatomical data

Pablo Fernandez-Gonzalez PABLO.FERNANDEZGONZ@FI.UPM.ES
Technical University of Madrid, 28223, Campus de Montegancedo, Boadilla del Monte, Madrid

Pedro Larranaga PEDRO.LARRANAGA@FI.UPM.ES
Technical University of Madrid, 28223, Campus de Montegancedo, Boadilla del Monte, Madrid

Concha Bielza MCBIELZA@FI.UPM.ES
Technical University of Madrid, 28223, Campus de Montegancedo, Boadilla del Monte, Madrid

Abstract

This study explores the benefits of using multidimensional classification. It introduces a novel classifier, the class bridge decomposable multidimensional Gaussian classifier (CB-MGC) within the application of multiple state-of-the-art machine learning techniques to the Neuromorpho dataset. We formulate a supervised classification problem for predicting specie, gender, level one cell type, level two cell type, development stage and area of the neocortex based of a set of morphological features extracted from a neuron. Additionally, we show a performance comparative between the different classifiers and discuss its results.

1. Introduction

It has been documented that neurons in different animals, developmental stage and different genders tend to present some morphologically distinctive features (Jacobs et al., 2014). The analyzed dataset (Ascoli et al., 2007) consists of 596 neurons of five different species over which a very detailed measurement, consisting of 185 features has been performed. Some of these features include, for example, the total surface of the soma, the number of branches of the dendrites, the length of the branches (minimum and maximal length as well) and many others. Using this dataset we can build a supervised classification problem with the aim of, under the presence of similar measurements of a neuron, determining the specie (rat, human, mouse and elephant), the gender (male or female), the cell type level one (if the cell is considered principal cell or interneuron), the cell type level two (pyramidal, stella, neurogliaform, containing cell, basket or bitufted), the development stage (adult, young, neonate or old) and the area of the neocortex (motor, somatosensory, fronto-insula, anterior cirgulate, entorhinal, occipital lobe, multiple, frontal lobe, insular cortex, frontopolar, postcentral gyri, precentral gyri, media prefrontal cortex and perirhinal). This problem is multidimensional, that is, there is more than one class to predict and the classes are not restricted to binary values. More concretely, the cardinality of the total space is 5376 possible label combinations. We will approach this problem using a set of implemented multidimensional classifiers and compare their data classification performances.

Moreover, for this work we have developed a classifier called class brigdge decomposable Multidimensional gaussian classifier (CB- MGC) that belongs to the category of Bayesian network classifiers. Our classifier explores beyond the common restrictions (one class, tree-like structures in the class variables, ...etc) in network structures for classification in Bayesian networks and also the handles the simultaneously variables of continuous and discrete nature for the multidimensional case. Its influenced by the works of (Pérez et al., 2006) and (Borchani et al., 2010). The strengths of our classifier are the capability to capture dependencies between the class variables and its ability to handle feature variables of continuous nature straightforwardly, without the need to discretize the data. The inspection of the 185 selected features suggested continuous variables that tend to distribute accordingly to Gaussian distributions. This, while clearly not sufficient, did encourage us to test the performance of this classifier and use it for this problem.

ICML Workshop on Statistics, Machine Learning and Neuroscience

2. Multidimensional Gaussian Network Classifier

A Bayesian network over a set of random variables $\chi = \{X_1, X_2, \ldots, X_n\}$ is a 2-tuple $\mathcal{B} = (\mathcal{G}, \Theta)$ where $\mathcal{G} = (V, A)$ is a directed acyclc 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. Vertices represent the variables in χ and arcs represent direct dependency 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 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 particular case of 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 random variables where vector χ_c is assumed to be jointly distributed as a multidimensional Gaussian distribution $\mathcal{N}(\mathbf{m}, \sum)$ (where $\mathbf{m}$ is a vector of means and $\sum$ is the covariance matrix of the variables in χ_c). They are referred to as the set of feature variables and the set of class variables, respectively. MGNCs are additionally constrained to satisfy $\mathbf{Pa}(C_i) \cap \chi_d = \emptyset$. Multidimensional classifiers have been studied intially in (Van Der Gaag et al., 2006), extended in (Bielza et al., 2011) and (Borchani et al., 2010).

In concordance with the literature, we additionally address a MGNC 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 its 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 Bayesian network as is denoted as $\mathcal{G}_{CX} = (V, A_{CX})$

In this work we restrict ourselves to datasets with no missing data. For this case, classification using a 0-1 loss function is achieved by computing the most probable explanation (MPE) given a data instance. That is, for an instance of the feature variables $\mathbf{x} = \{x_1, \ldots, x_m\}$ we need to obtain:

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

That is, to assign the class that maximizes the posterior probability of the MGNC. When computing MPE in a MGC, its possible to use Equation (1) to compute the MPE 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 variable in a BN and the feature nodes $p(x_i|\mathbf{pa}(x_i)$ follow a Gaussian distribution $\mathcal{N}(m_i, v_i)$ where

- $m_i = \mu_{i|pc_i} + \sum_{j=1}^{n_i} \beta_{ij|pc_i}(x_j - \mu_{j|pc_i})$
- $v_i = \frac{\left|\sum_{X_i, PX_{i|pc_i}}\right|}{\left|\sum_{PX_{i|pc_i}}\right|}$

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 GN possess several desired properties such as the less demanding number of parameters to model a continuous distribution ($\mathcal{O}\left(\prod_{i=1}^{n} r_i\right)$ where r_i are the cardinalities of the variables vs. $\mathcal{O}(n^2 r)$) and the possibility to compute them independently from the structure of the GN (Geiger & Heckerman, 1994). However, the computation of the MPE concerns only the class variables, that is, the discrete part of the network, and therefore no complexity alleviation is directly derived from this extension.

In order to tackle this problem, we now consider a subtype of MGNC called CB-decomposable MGC, extending previous works (Borchani et al., 2010) with multidimensional classifiers with discrete feature variables. A MGNC is a CB-decomposable MGC if it satisfies the following two propositions:

- $\mathcal{G}_C \cup \mathcal{G}_{CX}$ 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 MGC is transformed into

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

where $c^{\downarrow V_{C_{i\in I_i}}}$ is the projection of the vector 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 needs of the classification problem.

We implement for this work a CB-decomponsable MGC classifier and apply it to the neuromorpho dataset. Our learning algorithm is inspired in the work of (Borchani et al., 2010) extending it in the direction of continuous features following a Gaussian distribution.

3. Algorithm for learning the structure of a CB-Decomposable MGNC

Our learning algorithm can be characterized as a 3-step wrapper learning algorithm with a greedy forward search approach. That is, we initialize the arcs of the three different subgraphs to the empty set $A_C = \emptyset, A_X = \emptyset$ and $A_{XC} = \emptyset$, obtaining an initial network with no arcs and all nodes present. Then, we consider the addition of arcs to the different parts of the network judging their contributions using the global accuracy criteria, aiming to obtain a sufficiently good local or global optimal structure.

3.1. Learning the brigde subgraph

The algorithm starts by learning the bridge subgraph. For this we learn a Gaussian selective naive Bayes for each class variable $C_i, i = 1, \ldots, n$. We first select a class variable C_i and compute the mutual information $I(X_j, C_i), j = 1, \ldots, m$ between each of the feature variables X_j and the selected class variable. This provides us with an initial ranking R_{ij} for a feature variable j w.r.t. a class variable i. Then we sort the features by $I(X_j, C_i)$ in descending order, and add arcs to obtain the estimated accuracy for the subproblem of classifying only that single class. We then pick the structure that maximizes the accuracy. Finally, we eliminate shared children in order to obtain an initial CB-decomposable MBC structure with the maximum number of r maximal connected components. In order to do this, we first compare the ranks of each pair of shared variables w.r.t. their parent class variables, the one with the highest ranked is kept and the other arc eliminated. If the ranks are equal, we compare the accuracies of the Gaussian naive Bayes and keep the arc in the NB that presents the highest. The algorithm is depicted as follows:

1. **for** i = 1 **to** $n - m$ **do**
   - 1.1 Select class variable C_i,
   - 1.2 Perform m mutual information computations for all the feature variables $I(X_j, C_i)$ and use the produced values to sort the feature variables from that with the highest mutual information $X_{1,e}$ to that with the lowest $X_{m,f}$ (e, f being the original subscripts).
   - 1.3 Obtain an initial accuracy a_0 by classifying all instances as the most frequent class label. Initialize a set of accuracies A_i as $A_i = \{a_0\}$. Initialize the set of feature as $\chi_i = \emptyset$
   - 1.4 **for** j = 1 **to** m **do**
     - 1.4.1 $\chi_i := \chi_i \cup X_{j,.}$
     - 1.4.2 Compute Accuracy a_j of the current naive Bayes classifier, $NB_i(C_i, \chi_i)$, with C_i as class and χ_i as its feature set.
     - 1.4.3 $A_i := A_i \cup a_j$
   - 1.5 Obtain the $NB_i(C_i, \chi_e)$ that satisfies $\max(A_i) = Acc(NB(C_i, \chi_e))$ (χ_e can be easily inferred from the position of the maximum value a_e in the A_i set)
2. Compare all the children of NB_i and for each pair $NB_a(C_a, \chi_e), NB_b(C_b, \chi_d)$ such that $\chi_e \cap \chi_d \neq \emptyset$ do
   - 2.1 Compare $R_{ak} = I(C_a, \chi_k)$ (where subscript k references an existing shared variable between both Naive Bayes) with $R_{bk} = I(C_b, \chi_k)$ and eliminate the arc from A_{CX} corresponding to the lowest value. **If** $R_{ak} = R_{bk}$ **do**
     - 2.1.1 Compare $\max(A_a)$ with $\max(A_b)$ and eliminate the arc from A_{CX} in the naive Bayes with the lowest value
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 we will define a threshold number t of permitted arc insertions due to the computational burden associated with the examination of all possible arc insertions. For each arc insertion between a pair of nodes X_i, X_j, we evaluate the

accuracy of inserting the arc in both directions and subsequently check if the highest resulting global accuracy improves the current global accuracy of the classifier. Notice here how arc insertions in the feature subgraph can be computed locally and we can use as a reference the accuracies obtained in the previous steps. More concretely, only the accuracy concerning the r-maximal connected component containing the variable that has gained a parent, needs to be recomputed.

3.3. Fusion of components

The last step of the algorithm explores the dependencies between class variables and attempts to merge the r maximal connected components. At a first, all arc insertions in both directions between all classes are considered. If there is accuracy improvements, the arc that increases the accuracy the most is selected and the number of r maximal connected components is reevaluated. When two maximal connected component are merged, the bridge subgraph is updated considering arc insertions in a greedy way and updating until no further accuracy improvement is detected. This process is repeated until only a single $(r = 1)$ maximal connected component remains or no accuracy gain is detected by modifying the class subgraph. The algorithm is depicted as follows:

1. Initialize AccuracyImprovement = true, $R_c = \{r_1, \ldots, r_n\}$ where each $r_i \in R_c$ is a set containing the class variables of the associated r maximal connected components.
2. **while** AccuracyImprovement and $|R_c| > 1$ **do**
   - 2.1 Evaluate all possible arc insertions in pairs $C_i \rightarrow C_j, C_i \leftarrow C_j$ where $i, j = 1, \ldots, n$ and $i \neq j$ and obtain classifier accuracies $Acc = \{a_1, \ldots, a_k\}, k = 1, \ldots n^2 - n$
   - 2.2 **do** $A_{g+1} = \max(Acc)$
   - 2.3 **if** $A_{g+1} > A_g$ **then**
     - 2.3.1 Given the arc $C_i \rightarrow C_j$ that produced the maximal accuracy, update all $r_i \in R_c$ that include C_i as $r_i = r_i \cup C_j$ and viceversa.
     - 2.3.2 **if** more than a single r_i was modified (two maximum) **then** eliminate all but one of the modified r_i. This also implies that $|R_c| := |R_c| - 1$
     - 2.3.3 update $\mathcal{G}_C$
     - 2.3.4 **do**
       - 2.3.4.1 insert random arc and obtain $A_{r_i g+1}$
     - 2.3.5 **while** $A_{r_i g+1} > A_{r_i g}$
     - 2.3.6 Update $\mathcal{G}_{CX}$
   - 2.4 **else** Accuracy Improvement = false
3. Return the obtained CB-MGC

The computation alleviation is also produced by sufficient local computations of MPE. More concretely, each possible arc between class variables can be computed locally. Also, in the case of the bridge updating, only the concerned merged maximal connected component needs to be reevaluated.

4. Experimental study

In this study we use our novel classifier together with a representative subset of the classifiers provided by the MEKA tool. MEKA is an extension of the well-known WEKA (Hall et al., 2009) classification software that allows us to work with multiple class variables and multiple labels per class. After a preprocessing and reformatting step, that yielded a suitable input for the MEKA tool and our classifier, we have produced a table that summarizes all found results (Table 1). All the classifier accuracies in MEKA were obtained by a 10-fold cross validation. We have also produced a second table (Table 2) that shows, for the same classifiers, other two performance measures: Hamming score (the accuracy for each label averaged over all labels), that is expressed as:

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

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. and exact match (the accuracy of each example where all label values must match exactly for an example to be correct), that is expressed as:

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

Where $\delta_{T_i}^{P^i}$ is a function that outputs 1 if $T_i = P_i$ and 0 otherwise.

The classifiers are named, top to bottom: binary relevance classifier (BN), pruned C4 chain classifier (CC), classifier dependency network (CDN), classifier dependency trellis (CDT), deep back propagation neural network (DBPNN), hierarchical label sets (HASEL), label combination/labelpowerset (LC), majority labelset (ML), Montecarlo classifier chains (MCC), random k-label pruned sets (RAkEL) and rank + threshold (RT), class bridge decomposable multidimensional gaussian classifier (CB- MGC). All of these classifiers, but the latest, are explained in a recent review (Zhang & Zhou, 2014).

The best and the worst performing algotirhms are, respectively, the chain classifier (both with and without montecarlo optimization) and the deep back propagation neuronal

network. The first has the highest Hamming score and exact match of all competitors and scores the best accuracy in cell type level 2 and neocortex. The second has the worst exact match and performs significantly lower than the others in classes cell type level 2, development and neocortex. In the overall scenario, neocortex seems to be the hardest of all classes, followed by specie, as it is observed a generalized tendency to score lower in accuracy in both classes. This may be, specially in the case of the neocortex class (as it has 15 possible values), partially explained by the elevated number of labels and the low frequencies of the instances with those labels in the data. On the other hand, the easiest class seems to be gender, closely followed by cell type level 1. These two classes show an average accuracy superior to 90% correctly classified instances.

The CB-MGC shows acceptable performances ranking on average better than half of the classifiers of table1 for indidivual accuracies. In the case of the Hamming score and exact match, simila results were obtained. Due to time constrains, our classifier was not evaluated using a 10-fold cross validation procedure and therefore, its accuracies are those directly obtained by comparing the predicted classes and the real classes in our dataset. The learned structure of our Gaussian network connects 78 features to the class "species", 2 features to "development", 3 features to "cell type level one" and one feature to "neocortex". This may be interpreted as the specie being the most discriminant factor of the morphology of a neuron when compared to the other classes. The resulting network connected all five classes, showing the stronger relationships (where its "strength" was approximated by an *a posteriori* mutual information computation between the classes) between "species", "development" and "neocortex" together with "gender", "development" and "neocortex".

Table 1. Classification accuracies for the Neuromorpho neurons dataset.

Classifier	Species	Gender	Cell type level one	Cell type level two	Development	Neocortex
BR	0.67 ± 0.093	0.968 ± 0.022	0.988 ± 0.011	0.886 ± 0.052	0.719 ± 0.028	0.278 ± 0.066
CC	0.911 ± 0.039	0.954 ± 0.026	0.985 ± 0.013	0.959 ± 0.022	0.835 ± 0.046	0.721 ± 0.058
CDN	0.542 ± 0.113	0.937 ± 0.038	0.912 ± 0.041	0.886 ± 0.051	0.486 ± 0.113	0.402 ± 0.092
CDT	0.785 ± 0.195	0.952 ± 0.028	0.912 ± 0.041	0.884 ± 0.051	0.829 ± 0.052	0.553 ± 0.095
DBPNN	0.198 ± 0.038	0.932 ± 0.042	0.912 ± 0.041	0.835 ± 0.064	0.363 ± 0.068	0.129 ± 0.046
HASEL	0.651 ± 0.101	0.968 ± 0.022	0.988 ± 0.011	0.867 ± 0.059	0.678 ± 0.057	0.267 ± 0.061
LC	0.651 ± 0.089	0.964 ± 0.022	0.978 ± 0.016	0.896 ± 0.057	0.712 ± 0.052	0.284 ± 0.071
ML	0.194 ± 0.071	0.932 ± 0.042	0.912 ± 0.041	0.835 ± 0.064	0.446 ± 0.076	0.180 ± 0.077
MCC	0.911 ± 0.039	0.954 ± 0.026	0.985 ± 0.013	0.959 ± 0.022	0.835 ± 0.046	0.721 ± 0.058
RAkEL	0.619 ± 0.028	0.884 ± 0.035	0.866 ± 0.029	0.799 ± 0.054	0.283 ± 0.077	0.226 ± 0.076
RT	0.492 ± 0.077	0.932 ± 0.042	0.912 ± 0.041	0.835 ± 0.064	0.446 ± 0.076	0.180 ± 0.077
CB-MGC	0.7248	0.9316	0.9128	0.8325	0.5915	0.4068

Table 2. Hamming score and exact match values for the studied classifiers

CLASSIFIER	HAMMING SCORE	EXACT MATCH
BR	0.751 ± 0.013	0.097 ± 0.04
CC	0.894 ± 0.023	0.642 ± 0.058
CDN	0.694 ± 0.055	0.368 ± 0.09
CDT	0.819 ± 0.051	0.434 ± 0.088
DBPNN	0.561 ± 0.019	0 ± 0
HASEL	0.737 ± 0.021	0.097 ± 0.036
LC	0.748 ± 0.012	0.114 ± 0.037
ML	0.583 ± 0.041	0.153 ± 0.075
MCC	0.894 ± 0.23	0.642 ± 0.058
RAKEL	0.619 ± 0.028	0.121 ± 0.068
RT	0.633 ± 0.024	0.027 ± 0.021
CB-MGC	0.8014	0.2496

5. Discussion

We considered that the problem of correctly classifying the neurons provided their morphological description has met with very acceptable accuracies in more than half of the classes considered, which seems a promising result. Future research could be aimed to test the set of features with feature subset selection techniques and compared the results for different classifiers with the ones obtained in the table. Also it is of a very practical interest to consider the structure of dependencies that exists between the classes. In table 1, the differences between the binary relevance methods and the chain classifiers, specially regarding the neocortex class, could be interpreted as an indicator of existing dependencies between classes.

From the obtained results we can conclude that, for this problem, our classifier meets levels of individual accuracies, exact match and Hamming score within the state-of-the-art exigences for this dataset, and hence is susceptible to be used as a benchmarking algorithm in the future.

Nevertheless, since the possible structures to be learned with a Bayesian network of this number of variables is quite large, other learning strategies may be considered for the same dataset as other local optima, perhaps with the capability to improve all scores, are expected to be found. Additionally, it should be noted that in general, more interdependence between classes carries with it higher data size requirements, together with the cardinality of possible distinctive values in each class. Thus, its expected that data scarcity decreases performance more in classifiers that aim to capture a complete dependency structure between class variables, than in others. The addressing of this considerations would, to our belief, increase our classifiers performance to more promising results.

The two main motivations for this article lies are the study and comprehension of the morphological details of the neurons, (for which our algorithm and multidimensional classification could be used to assist researchers and laboratories that lack some of the information gathered by the class variables) and the introduction an application of our algorithm to a real dataset in the context of multidimensional classification.

6. Acknowledgements

This work has been partially supported by the Spanish Ministry of Economy and Competitiveness through the Cajal Blue Brain (C080020-09; the Spanish partner of the Blue Brain initiative from EPFL) and TIN2013-41592-P projects, by the Regional Government of Madrid through the S2013/ICE-2845-CASI-CAM-CM project.

References

Ascoli, G. A., E., Donohue D., and Halavi, M. Neuromorpho.org: a central resource for neuronal morphologies. *The Journal of Neuroscience*, 2007. ISSN 92479251.

Bielza, C., Li, G.i, and Larrañaga, P. Multi-dimensional classification with bayesian networks. *International Journal of Approximate Reasoning*, 52(6):705–727, 2011.

Borchani, H., Bielza, C., and Larrañaga, P. Learning cb-decomposable multi-dimensional Bayesian network classifiers. In *Proceedings of the 5th European Workshop on Probabilistic Graphical Models (PGM10)*, pp. 25–32, 2010.

Geiger, D. and 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.

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., and Witten, Ian H. The WEKA data mining software: An update; SIGKDD explorations. 11, 2009.

Jacobs, B., Johnson, N., Wahl, D., Schall, M., Maseko, B C., Lewandowski, A., Raghanti, MA., Wicinski, B., Butti, C., Hopkins, W., Bertelsen, MF., Walsh, T., Roberts, JR., Reep, R., Hof, PR., Sherwood, C. and Manger, Pl. Comparative neuronal morphology of the cerebellar cortex in afrotherians, carnivores, cetartiodactyls, and primates. *Frontiers in Neuroanatomy*, 8(24), 2014.

Pérez, A., Larrañaga, P., and 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. ISSN 0888-613X.

Van Der Gaag, L. and De Waal, P. Multi-dimensional bayesian network classifiers. In *Probabilistic graphical models*, pp. 107–114. Prague, 2006.

Zhang, M-L., and Zhou, Z-H. A review on multi-label learning algorithms. *Knowledge and Data Engineering, IEEE Transactions on*, 26(8):1819–1837, Aug 2014. ISSN 1041-4347. doi: 10.1109/TKDE.2013.39.